



UNIVERSIDAD TÉCNICA NACIONAL

VICERRECTORÍA DE INVESTIGACIÓN Y POSGRADO

CENTRO DE FORMACIÓN PEDAGÓGICA Y TECNOLOGÍA EDUCATIVA

MAESTRÍA EN ENTORNOS VIRTUALES DE APRENDIZAJE

PROYECTO DE INTERVENCIÓN

TÍTULO DEL PROYECTO:

Virtualización del Curso Programación I de la Carrera Ingeniería Electrónica

PREPARADO POR: Jonathan Arias Román

TUTOR DEL PROYECTO: Mariela Delauro

2018

Tabla de Contenidos

Resumen Técnico.....	4
1 Planteamiento del problema.....	6
1.1 El problema.....	6
1.2 La justificación del problema.....	6
1.3 El contexto del problema.....	8
2 Prospectiva.....	13
3 Propuesta pedagógica.....	14
4 Objetivos: General y Específicos.....	16
4.1 Objetivo General.....	16
4.2 Objetivos Específicos.....	16
5 Resultados esperados.....	17
6 Aspectos operativos.....	18
7 Evaluación y Seguimiento.....	21
7.1 Evaluación.....	21
7.2 Seguimiento.....	23
8 Cronograma para ejecución del proyecto.....	24
9 Presupuesto.....	26
10 Bibliografía.....	27
DESARROLLO DEL PROYECTO.....	28
1. Nombre del curso virtual.....	29
2. Selección y Justificación de las herramientas tecnológicas.....	29
3. Planificación de las clases.....	31
4. Redacción de las clases.....	48
5. Captura de pantalla de las clases.....	59
DOCUMENTOS ELABORADOS.....	67
Guía Didáctica.....	1
1 Fundamentación de la materia.....	1
2 Objetivos Generales y Específicos.....	1
3 Contenidos.....	2

4 Metodología de Trabajo.....	3
5 Evaluación de los aprendizajes.....	4
6 Cronograma de Trabajo.....	6
7 Presentación del tutor.....	7
Módulo #1 Estructuras de Control y Diagramas de Flujo.....	1
Estructuras de Control: Parte I.....	3
Iniciamos nuestro camino.....	3
Instrucción de selección if.....	5
Instrucción de selección if...else.....	7
Instrucción de repetición while.....	8
Estructuras de Control: Parte II.....	10
Instrucción de repetición while.....	10
Instrucción de repetición for.....	12
Instrucción de repetición do...while.....	15
Diagramas de Flujo.....	17
Ventajas de utilizar diagramas de flujo.....	17
Símbolos utilizados en diagramas de flujo.....	18
Aplicaciones para diseño de Diagramas de Flujo.....	19
PSeInt.....	19
Raptor.....	19
SmartDraw.....	19
Bibliografía.....	20
Conclusiones.....	96

Resumen Técnico

El presente documento tiene como objetivo mostrar todo el proceso de desarrollo de la virtualización en la plataforma Moodle del curso Programación I de la carrera Ingeniería Electrónica de la Universidad Técnica Nacional Sede Central.

Se parte de la búsqueda, análisis y justificación del problema para el planteamiento de una solución virtual que ayude a la comunidad estudiantil a mejorar las condiciones de acceso a la educación superior, para tratar de mejorar la deserción y el rendimiento académico.

Se desarrolla todo el proceso logístico, pedagógico y tecnológico de diseño e implementación del curso en modalidad virtual, realizando la planificación en el tiempo para la implementación del proyecto hasta obtener un producto final para ser utilizado en el curso de forma virtual.

Por último se muestran los resultados obtenidos en el desarrollo del proyecto de virtualización del curso con el objetivo de mostrar los resultados obtenidos, generar evidencias de la conclusión del proyecto y validar el cumplimiento del objetivo del proyecto de intervención de maestría.

PROPUESTA DEL PROYECTO

1 Planteamiento del problema

1.1 El problema

En la Universidad Técnica Nacional Sede Central ubicada en la provincia de Alajuela Costa Rica, se imparte la carrera de Ingeniería Electrónica con salida lateral de bachillerato universitario, donde existen tres franjas horarias para los diferentes perfiles de estudiantes con los cuales cuenta la universidad, por lo que en este caso el problema a tratar se encuentra ubicado en la franja nocturna para estudiantes que laboran tiempo completo y solo cuentan con tiempo disponible en la franja nocturna para poder formarse en una carrera universitaria, se les dificulta poder desplazarse o asistir de forma presencial a las lecciones afectando el rendimiento académico.

1.2 La justificación del problema

En la Universidad Técnica Nacional Sede Central ubicada en la provincia de Alajuela Costa Rica, se imparte la carrera de Ingeniería Electrónica con salida lateral de bachillerato universitario, donde existen tres franjas horarias para los diferentes perfiles de estudiantes con los cuales cuenta la universidad, en este caso el problema a tratar se encuentra ubicado en la franja nocturna para estudiantes que laboran tiempo completo y solo cuentan con tiempo disponible en la franja nocturna para poder formarse en una carrera universitaria.

Dado este perfil de estudiante con el cual trabaja la carrera de ingeniería en electrónica en su franja nocturna, es que se presentan problemas de ausentismo por parte de la población, junto con problemas de abandono del curso por situaciones laborales de fuerza mayor que amerita que los estudiantes deban tomar la difícil decisión entre mantener su trabajo y poder llevar el sustento a sus familias o llevar el curso, por lo que se presenta la dificultad de asistencia al mismo y por lo tanto los problemas en rendimiento académico junto con la dificultad de ponerse al día con el curso y el entendimiento de los contenidos del mismo.

Con la integración de un Entorno Virtual de Aprendizaje en específico a los cursos de Programación I y Programación II que se imparten actualmente en la franja nocturna, traerá beneficios significativos a los estudiantes, ya que pese al problema de ausentismo a lecciones por el tema de disponibilidad laboral junto con la problemática de mantener al día los contenidos del curso y el correcto entendimiento de los mismos, con la ayuda del Entorno Virtual de Aprendizaje los estudiantes contarán desde el día uno con todo el cronograma de actividades, evaluaciones y lecturas del curso, junto con los materiales de estudio y los enunciados de las evaluaciones virtuales, el estudiante puede organizar su tiempo y tener acceso a toda esta información en cualquier momento y tomar notas y dudas sobre los contenidos, logrando un mejor aprovechamiento de la clase presencial para evacuar dudas sobre los contenidos y con ello una expectativa de mejor rendimiento académico en el proceso de aprendizaje.

Cabe mencionar que esta primera etapa del proyecto se impartirá en modalidad mixta, ya que al ser una carrera cien por ciento presencial al igual que sus cursos, se implementará una alternativa híbrida como acompañamiento al curso presencial para que los estudiantes tengan otra alternativa paralela complementaria al curso principal y con ello en caso de presentar los problemas antes mencionados puedan tener un apoyo por medio de la plataforma virtual y realizar la mayoría de las evaluaciones que originalmente son presenciales por medio del entorno virtual de aprendizaje propuesto en este proyecto.

Cabe mencionar que se optó por este problema por ser una problemática general de toda la población de la carrera de electrónica en su franja nocturna, que pese a ser un proyecto que solo se va a implementar en los cursos de Programación I y Programación II, es de gran importancia dado que al ser los primeros en romper el hielo en la implementación de esta metodología y los pioneros en desarrollar el proyecto a los cursos de electrónica, esto abrirá las puertas para que futuros compañeros docentes y la escuela de electrónica en general valoren esta alternativa y puedan ir incursionando en el mundo de los Entornos Virtuales de Aprendizaje poco a poco con el trabajo a desarrollar.

1.3 El contexto del problema

La problemática presentada está centrada en los estudiantes que laboran tiempo completo y solo cuentan con tiempo disponible en la franja nocturna para poder estudiar y formarse en una carrera universitaria. Con este perfil de estudiante con el cual trabaja la carrera de ingeniería en electrónica en su franja nocturna, es que se presentan problemas de ausentismo por parte de la población, junto con problemas de abandono de cursos por situaciones laborales de fuerza mayor que amerita que los estudiantes deban decidir entre conservar su trabajo o seguir en el curso, presentándose la dificultad de asistencia al mismo y por lo tanto los problemas de rendimiento académico junto con la dificultad de ponerse al día con los contenidos del curso y el entendimiento de los mismo.

Como se mencionó anteriormente la problemática presentada no es una problemática específica de un curso, sino una problemática presente en toda la población de la franja nocturna, donde en este caso se está proponiendo un proyecto que complementa de forma virtual el curso de Programación I de la carrera de Ingeniería Electrónica.

Las principales características de la asignatura se mencionan a continuación:

Curso: Programación I

Código: IEL-721

Nivel: VII

Naturaleza del Curso: Teórico-Práctico

Créditos: 2

Modalidad: Cuatrimestral

Horas Presenciales / Semana: 4

Tiempo de Estudio Independiente Por Semana: 2 Horas, 26 Minutos

Requisitos: Ninguno

Co-Requisito: Ninguno

Evaluación del Curso

Rubro	Porcentaje
Primera Prueba	25%
Segunda Prueba	25%
Proyectos Cortos	25%
Proyecto Programado	25%
Total	100%

Este curso pretende que los aprendientes apliquen las nociones y los elementos básicos de un lenguaje de programación, orientado principalmente a la utilización del lenguaje de programación C++; mediante el diseño de algoritmos y la utilización de estructuras de decisión y control.

Las principales temáticas a desarrollar están enfocadas en los aspectos básicos de la programación, utilizando diferentes herramientas que permitirán conocer y familiarizarse con la aplicación, para posteriormente poder desarrollar soluciones diversas. La programación es un elemento fundamental en la educación de un Ingeniero en Electrónica, debido a que muchos dispositivos y elementos electrónicos digitales se programan con algún tipo de lenguaje de programación.

Se utiliza una metodología participativa, con interacción docente-estudiante, a través de la realización de prácticas de programación que integren el uso de algún protocolo de comunicación a través de un puerto de la PC para que se comunique con el mundo externo.

El profesor explica los conceptos y nociones básicas de la programación, haciendo uso de equipo audiovisual, con el fin de demostrar las funciones que va explicando, desarrollo de prácticas y proyectos cortos en cada lección y un proyecto de programación final.

En el curso se establece una participación activa de los estudiantes para que de esta forma, desarrollen un proceso de aprendizaje integral y se logren aplicar los conocimientos, habilidades y actitudes en el ámbito profesional.

Se proyecta que cuando se implemente el Entorno Virtual de Aprendizaje como acompañamiento al curso de Programación I, los estudiantes que presenten problemas de ausentismo por situaciones laborales, no se vean afectados por los contenidos perdidos por la inasistencia a clases, con ayuda del entorno virtual puedan llevar un seguimiento al día de la materia del curso desde cualquier lugar y en el momento en el cual tengan la disponibilidad para poder estudiar los contenidos que fueron vistos en clases. Por otro lado el tema de las actividades programadas que se realizan en clases se lograría solventar realizándose las mismas por medio del entorno virtual en una ventana de tiempo prudente en la cual el estudiante pueda realizar la evaluación desde cualquier lugar y en el momento donde cuente con el tiempo suficiente para realizar la evaluación correspondiente. Todo esto con el objetivo final de brindarle las herramientas necesarias al estudiante para no verse afectado por las situaciones antes expuestas y con ello que se logre mejorar el rendimiento académico del curso junto con el porcentaje de aprobación.

Modelo Educativo

La UTN es una institución estatal de educación superior universitaria cuyo fin es dar atención a las necesidades de formación técnica que requiere el país. Su misión es “brindar una educación integral en el marco de la moderna sociedad del conocimiento, centrando su acción académica en el área científica, técnica y tecnológica, en la investigación de alta calidad y en la innovación como elemento fundamental para el desarrollo humano con responsabilidad ambiental, en articulación con los sectores productivos de la sociedad” (Plan Institucional de Desarrollo Estratégico, 2011-2021).

En el cumplimiento de esta apoteósica tarea social, se genera una propuesta educativa integral basada en las ciencias de la complejidad y que supone el reconocimiento del mundo como una red de relaciones entre las distintas partes de un todo global y en las que todos aprendemos constantemente. (UTN, 2017)

Nuestro modelo educativo plantea estrategias y prácticas pedagógicas centradas en el aprendizaje, así como una gestión de formación holística en la que las habilidades sociales se complementan con las habilidades y conocimientos específicos del área de trabajo. Desde esta visión holística, las prácticas educativas en la UTN deben encontrar su origen en el pensamiento crítico y creativo, en el desarrollo de líderes participativos comprometidos con la sociedad, cuya formación es contextualizada a partir de estudios interdisciplinarios, de experiencias transdisciplinarias y de proyectos que faciliten la comprensión, la reflexión y las respuestas innovadoras. (UTN, 2017)

El proceso educativo en la UTN supone, entonces, además de la integración de la enseñanza y el aprendizaje, la integración sistémica de todos los actores de dicho proceso; para ello requiere que desde la gestión educativa, se respalde y se apoye al aprendiente, ofreciéndole nuevas alternativas, posibilidades y retos que le estimulen en su búsqueda de información y en la construcción de nuevos conocimientos, a través de estrategias de aprendizaje colaborativo, interaprendizaje y autogestión del aprendizaje. Planteamos un diseño curricular interdisciplinar e integrador que, en el marco de la flexibilidad y la transformación y, en la consecución de perfiles profesionales que respondan a las necesidades del mundo productivo y al continuo avance de la tecnología como motores de desarrollo social, valore tanto el saber conocer y saber hacer como el saber ser y saber convivir; un currículo que permita la producción de conocimientos, tanto en los estudiantes como en los docentes. De igual manera, la gestión curricular está orientada al desarrollo humano y de competencias profesionales y personales, en el marco de la sociedad del conocimiento. (UTN, 2017)

Proponemos un proceso educativo que, desde toda la gestión institucional, cultive una conciencia crítica en las dimensiones social, cognoscitiva, científica, ecológica, económica, tecnológica, espiritual y política, que además fomente y coadyuve en la formación de profesionales que contemplan en sus atisbos, un alto sentido de responsabilidad de sí mismos y con los demás. La formación integral desde un enfoque filosófico, humanista y científico, se sitúa como elemento nuclear de su gestión educativa y va dirigida a crear circunstancias cognitivas en las cuales los aprendientes logren una adecuada formación para

la vida, que integre lo técnico, lo humano, la innovación y el desarrollo, con el objetivo de fomentar la investigación, la implementación de nuevas tecnologías, el desarrollo sostenible, la conservación del ambiente y el emprendedurismo. (UTN, 2017)

Desde una perspectiva innovadora, entonces, la oferta académica de la UTN apuesta por el desarrollo de competencias profesionales y laborales. Así, la formación académica y profesional se distingue de otros centros de educación superior estatal en tanto el diseño de todas sus carreras procura salidas laterales en niveles de pregrado, con un perfil ocupacional y técnico claro que le facilita al graduado su inserción laboral y su vinculación con los sectores productivos del país, mismos que dan paso a un prolífico desarrollo y crecimiento académico institucional. (UTN, 2017)

Por otro lado la Universidad Técnica Nacional trabaja fuertemente con su nuevo programa de Maestría en Entornos Virtuales de Aprendizaje (UTN impartirá primera maestría en Entornos Virtuales de Aprendizaje) lo cual demuestra el interés y enfoque que la universidad visualiza en este modelo de educación para la educación a corto plazo.

2 Prospectiva

En dos años, el curso Programación I de la carrera Ingeniería Electrónica impartido por la Universidad Técnica Nacional sede central se ofrecerá en un entorno virtual de aprendizaje y será cursado a distancia por veinte estudiantes. Se trabajarán prácticas de aprendizaje basadas en un modelo de aprendizaje virtual, con la ayuda de la plataforma virtual Moodle, materiales didácticos digitales complementarios a cada unidad didáctica del curso, con un tutor como mínimo capacitado en el manejo de Entornos Virtuales de Aprendizaje y con la asesoría del Centro de Formación Pedagógica y Tecnología Educativa de la Universidad Técnica Nacional como ente regulador dentro del centro universitario.

El curso se sostendrá cien por ciento por medio de un Entorno Virtual de Aprendizaje el cual puede ser consultado por los estudiantes en cualquier momento, ya que la totalidad del curso estará disponible en el entorno virtual junto con la totalidad de las evaluaciones del mismo. El centro universitario en específico la carrera de Ingeniería Electrónica adoptará por completo la integración del Entorno Virtual de Aprendizaje en la totalidad de los cursos de programación, algunos otros relacionados directamente con el área de programación y sistemas digitales, con la salvedad que existen algunos cursos que son prácticos y siempre es necesaria la presencialidad de los estudiantes, pero con el entorno virtual como acompañamiento fundamental en la metodología de los cursos de la carrera de electrónica.

3 Propuesta pedagógica

Se plantea el desarrollo de un Entorno Virtual de Aprendizaje para dar solución a un problema educativo presente en la carrera de Ingeniería Electrónica de la Universidad Técnica Nacional Sede Central Alajuela en su franja horaria nocturna, pretendiéndose con el apoyo de tecnologías alternativas enfocadas a un entorno virtual, ayudar a mejorar el proceso de enseñanza-aprendizaje de los estudiantes del curso Programación I, aplicando nuevas alternativas tecnológicas al proceso para tratar de mejorar la problemática presente por el tema de ausentismo por situaciones laborales de los estudiantes ajenas a sus manos.

Con este proyecto se pretende incentivar a los estudiantes a utilizar formas alternativas de aprendizaje virtual que motiven su creatividad y ganas de seguir aprendiendo con una herramienta tecnológica que les ayudará y desafiará a ser creativos en el proceso, desde una posición diferente a las clases tradicionales.

De esta manera se tratará que el proceso de enseñanza-aprendizaje sea más significativo para cada uno de los estudiantes indistintamente si estos faltaron a las lecciones por los problemas antes mencionados, logrando con esto un seguimiento más de cerca por parte del estudiante a los contenidos de la asignatura, las actividades de retroalimentación en el entorno virtual, disponibilidad de foro de dudas con todos los miembros del grupo y el desarrollo de las actividades de evaluación por completo en el entorno virtual de aprendizaje.

Por otro lado la Universidad Técnica Nacional como pionera en el país en ser la primera universidad estatal en impartir un posgrado totalmente en modalidad virtual, plantea que la Maestría en Entornos Virtuales de Aprendizaje, está fundamentada en las Tecnologías de Información y Comunicación como herramientas de trabajo académico para facilitar el aprendizaje, promoviendo el trabajo basado en las buenas prácticas de la educación a distancia con utilización intensiva de tecnologías de la información y la comunicación, tutorías proactivas, diseño didáctico de los materiales, campus virtual con todas las prestaciones adecuadas y utilización de recursos didácticos. (UTN, 2017)

Además la población meta del posgrado son: (UTN, 2017)

- Docentes y pedagogos de nivel medio o superior, que aspiren a desempeñar su labor profesional en entornos virtuales de aprendizaje.
- Técnicos auxiliares (programadores, diseñadores gráficos, diseñadores web, etc.) que trabajen o aspiren a trabajar en equipos multidisciplinares de producción de contenidos para educación mediada en entornos virtuales.
- Directivos de instituciones educativas que estén en procesos de pasaje de parte de su actividad formativa a la modalidad a distancia, utilizando entornos virtuales.
- Responsables del área de recursos humanos de las empresas que estén implementando proyectos de e-learning.

4 Objetivos: General y Específicos

4.1 Objetivo General

Transformar el entorno tradicional de aprendizaje del curso Programación I de la Carrera Ingeniería Electrónica de la Universidad Técnica Nacional Sede Central mediante la incorporación de un Entorno Virtual de Aprendizaje a las prácticas de enseñanza y aprendizaje, a fin de solucionar el problema de seguimiento al día de los contenidos del curso por situaciones laborales de fuerza mayor que imposibilitan a la población estudiantil asistir constantemente a las lecciones por situaciones laborales.

4.2 Objetivos Específicos

- Implementar un Entorno Virtual de Aprendizaje para el curso de Programación I
- Capacitar a los estudiantes en el uso del Entorno Virtual de Aprendizaje
- Migrar el cien por ciento de la evaluación del curso al entorno virtual

5 Resultados esperados

Al cabo de 2 años se habrá llegado a los siguientes resultados:

- Como mínimo un 70% de aprobación en el curso de Programación I, cada cuatrimestre son grupos de diez a quince estudiantes según matrícula.
- Implementación de la plataforma virtual Moodle como herramienta tecnológica de acompañamiento al curso presencial.
- Capacitación técnica cuatrimestral de los estudiantes del curso en el uso de la plataforma virtual.
- Desarrollo de material didáctico propio para el curso de Programación I. Se seguirá utilizando el libro de texto oficial del curso pero además se realizará material didáctico complementario para reforzar y aclarar los conceptos de mayor importancia, además de los conceptos que son típicos de difícil comprensión por lo estudiantes.
- Desarrollo de la evaluación del curso en un cien por ciento en el Entorno Virtual de Aprendizaje.

6 Aspectos operativos

Etapa 1

Para el curso de Programación I en modalidad acompañamiento por medio de un Entorno Virtual de Aprendizaje, se desarrollarán tres foros a lo largo de todo el cuatrimestre, con una duración de ocho días cada uno y distribuidos en la semana tres, siete y once.

Se mantendrá abierto a lo largo de todo el cuatrimestre un apartado de foro de dudas y consultas para la interacción docente-estudiantes y estudiantes-estudiantes que presenten dudas y estas sean aclaradas, compartidas y desarrolladas por los mismos estudiantes e interacción del docente con tres revisiones semanales del foro de consultas y dudas por parte del docente.

Un apartado de noticias importantes, que informará cualquier eventualidad del curso y recordatorios de evaluaciones, publicaciones de las mismas y temas que sean de gran relevancia por informar a los cursantes del entorno virtual.

Las tareas programadas que se asignen a lo largo de todo el cuatrimestre serán publicadas en el entorno virtual y la solución a las mismas por parte de los estudiantes deberá ser subida al entorno virtual de aprendizaje para su revisión.

Etapa 2

El curso de Programación I utiliza el libro de texto:

Deitel & Deitel. (2008). C++ Cómo Programar. Pearson Education. México

Como material didáctico oficial de estudio y consulta por parte del estudiante, donde semana a semana se asigna la(s) lectura(s) correspondiente(s) a los temas por tratar.

Se realizará material didáctico complementario por cada unidad que se evalúe en el curso, para reforzar y aclarar los conceptos de mayor importancia, además de los conceptos que son típicos de difícil comprensión por los estudiantes, para reforzar dichos conceptos y tratar

de obtener mejores resultados en el proceso de aprendizaje.

Por otro lado se deben realizar las tres consignas de los foros planteados previamente, con sus respectivos enunciados, indicaciones y objetivos a desarrollar en cada foro. Además digitalizar todas las tareas programadas del curso y tenerlas disponibles desde el día uno en el entorno virtual de aprendizaje. Además planificar las fechas de cada semana donde exista una tarea programada.

Etapa 3

El tutor(a) de un entorno virtual de aprendizaje debe estar en constante revisión del entorno virtual, para dar un correcto seguimiento a cada uno de los estudiantes del curso. En este curso el tutor será el mismo docente del curso presencial el cual revisará como mínimo tres veces por semana el foro de dudas y consultas, cuando se desarrollen los foros de evaluación el docente publicará el primer día la consigna junto con sus respectivas indicaciones, dejará pasar un par de días y hará una intervención para animar a los foristas que no han participado a que participen y a los que ya lo han hecho realizar una publicación general sobre la orientación que debe tomar el foro y a seguirlos animando a participar. En el día cuatro o cinco del foro intervendrá de nuevo en la dinámica del foro pero sobre todo prestará especial atención a los foristas que no hayan participado y les enviará un mensaje privado amigable invitándolos a participar en el foro en curso. Al concluir el día seis del foro enviará de nuevo un mensaje a los foristas que no hayan participado de forma amable pero alertando sobre su baja o nula participación en el foro e indicándole que debe participar del foro de forma obligatoria como parte de una asignación por evaluar.

Con base a las tareas programadas estas serán publicadas en la semana correspondiente y el docente mandará un correo global sobre la publicación de la tarea y fecha de entrega, además publicará en el apartado de noticias importantes que ha publicado la tarea con su respectiva fecha de entrega.

El docente semana a semana realizará un seguimiento de cada uno de los estudiantes para revisar su caminar por el entorno virtual y enviará correos amigables de recordatorio en caso que sea necesario con estudiantes que vayan dejando un poco atrás las evaluaciones y lecturas asignadas semana a semana.

Etapa 4

Con respecto a la parte administrativa del Entorno Virtual de Aprendizaje, el Centro de Formación Pedagógica y Tecnología Educativa de la Universidad Técnica Nacional es el encargado junto con el Departamento de Admisión y Registro de la Sede Central, de coordinar todo tema relacionado con la inscripción de alumnos, mantenimiento y funcionamiento del Entorno Virtual de Aprendizaje de la plataforma Moodle y cualquier soporte técnico necesario por parte de los docentes en esta área.

Para esta etapa del proyecto que solo se va a implementar como entorno de acompañamiento al curso presencial de Programación I, el docente es el encargado de la entrega de materiales y administración de materiales didácticos, publicación de las evaluaciones con su respectivo cronograma de actividades, seguimiento de los cursantes del entorno virtual y sobre todo el guía que orienta el camino a seguir por parte de los estudiantes.

7 Evaluación y Seguimiento

7.1 Evaluación

El siguiente apartado pretende plantear una serie de indicadores que sirvan como referencia para poder evaluar el desempeño del proyecto en el transcurso del tiempo de ejecución y una vez finalizado el mismo, evaluar los puntos de mayor dificultad en la ejecución del proyecto, los puntos fuertes que se pueden volver a utilizar, entre otros, para luego proponer mejoras que serán ejecutadas en las siguientes iteraciones del proyecto.

- Modelo pedagógico general

1) Integración del estudiantado al Entorno Virtual de Aprendizaje por medio de la plataforma Moodle facilitada por el Centro Pedagógico de la UTN, junto con la asignación de evaluaciones, actividades formativas y sumativas en la plataforma que hagan que el estudiante se familiarice e incorpore al EVA desde el inicio del curso.

2) Formación holística en el proceso de enseñanza-aprendizaje, integrando herramientas virtuales, tecnológicas, enseñanza virtual, valores, entre otros para que el perfil del estudiante egresado del EVA cumpla con el perfil integral planteado por la Universidad.

- Prácticas de aprendizaje y tecnologías

1) Aumento del rendimiento académico del estudiantado, con ayuda del EVA para un seguimiento más de cerca de la teoría del curso, asesoría docente a través del EVA, actualización constante de los contenidos para el seguimiento de la población estudiantil junto con la implementación de evaluaciones por medio del EVA y que los estudiantes no dependan de asistir al aula sino que se puedan acomodar según disponibilidad de su tiempo.

2) Inserción de tecnologías educativas al proceso de aprendizaje, con ayuda del Centro Pedagógico de la UTN que facilita y da soporte al EVA, aportes del docente a través del EVA que utilice las herramientas permitidas por el entorno virtual para obtener el máximo beneficio y facilidad de material didáctico para el estudiantado.

- Material didáctico

1) Material didáctico complementario por cada unidad, que el docente del curso desarrollará y subirá al EVA para que pueda ser descargado por los estudiantes en el momento deseado. Dicho material se subirá al EVA un día antes del inicio de la nueva unidad por estudiar.

2) Evaluaciones en el Entorno Virtual de Aprendizaje, foros de consulta, foros de actividades evaluativas, tareas con su respectiva consigna que deberán ser presentadas a través del EVA en las fechas establecidas.

- Tutoría

1) Revisión constante del Entorno Virtual de Aprendizaje por parte del docente, con accesos cada dos días para revisión de foros de consultas, foros de evaluación, revisión de tareas, entre otras actividades que se desarrollen en la plataforma, dado que con una revisión cada dos o tres días a la semana logrará cubrir las actividades y asignaciones del entorno.

2) Seguimiento y confección de materiales complementarios de evaluación (consignas), apertura de foros de dudas junto con cada actividad o consigna planteada, para que el estudiante tenga la oportunidad de comentar en los foros sus dudas, inquietudes, reciba retroalimentación a sus comentarios que lo ayuden en el desarrollo de la actividad en curso.

- Administración

1) Satisfacción de docentes y estudiantes con la administración del EVA y proceso de matrícula, que el docente cuente con las herramientas necesarias para el desarrollo de la clase virtual, los estudiantes no presenten problemas técnicos con la plataforma ni el proceso de matrícula, además una capacitación a los estudiantes en el manejo de la plataforma Moodle.

2) Entrega y administración de materiales didácticos, el docente es responsable de entregar los materiales didácticos un día antes del inicio de una nueva unidad por estudiar, junto con el seguimiento en caso de problemas técnicos presentados por los estudiantes.

7.2 Seguimiento

Inicialmente se tomará como referencia el cronograma de actividades por desarrollar en el proyecto, dándole seguimiento con las fechas de entrega planteadas de cada una de las tareas iniciales. El seguimiento del cronograma planteado es a nivel macro por lo que se necesita realizar una estrategia de seguimiento para la ejecución del proyecto a nivel micro.

- Se debe validar todo lo referente a la plataforma tecnológica para que funcione al cien por ciento tanto desde el lado del docente como del estudiante.

- Producción y entrega de materiales didácticos complementarios a tiempo y en las fechas establecidas.

- Revisión constante durante la ejecución del proyecto de los materiales didácticos en caso de dudas, entre otros que presenten los estudiantes y realizar los cambios correspondientes.

- Seguimiento de las consignas, actividades virtuales propuestas, foros entre otros y mantener una constante comunicación entre docente-estudiante según sea necesario.

- Mantener comunicación constante con el Centro Pedagógico de la UTN en caso de eventualidades, fallas, entre otros que ameriten intervención y ayuda directa por parte de ellos.
- Seguimiento, apoyo y comunicación directa con la dirección de la carrera de electrónica en caso de cualquier necesidad técnico o administrativo.
- Evaluación constante de los indicadores del proyecto y en caso de ser necesario realizar los cambios necesarios.
- Realizar muestreos periódicos en la ejecución del proyecto, como mínimo dos al inicio, desarrollo y cierre del proyecto.
- Realizar una análisis de lo observado, recomendaciones, conclusiones, mejoras necesarias, grandes aciertos para tener presentes para la próxima vez.

8 Cronograma para ejecución del proyecto

A continuación se presenta un cronograma estimado de la ejecución del proyecto.

Actividad	Tiempo de Ejecución																			
	Mes 1				Mes 2				Mes 3				Mes 4				Mes 5			
	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4	#1	#2	#3	#4
Puesta a punto de la tecnología	■	■	■	■	■	■	■	■												
Capacitación de los tutores			■	■	■	■	■	■	■	■	■	■	■	■						
Elaboración de los materiales									■	■	■	■	■	■	■	■	■	■	■	■
Organización del sistema administrativo			■	■	■	■	■	■	■	■										
Pruebas del sistema de la plataforma virtual											■	■	■	■	■	■	■	■		
Mercadeo y promoción de la nueva modalidad del curso	■	■			■	■			■	■			■	■			■	■		
Capacitación de estudiantes																	■	■	■	■

9 Presupuesto

Para el desarrollo del presente proyecto se cuenta con el laboratorio de computo, disponibilidad de la plataforma Moodle, asesoría por parte del Centro de Formación Pedagógica y Tecnología Educativa de la Universidad Técnica Nacional, pago de planilla a los docentes, capacitaciones en el uso de la plataforma virtual, docente capacitado y certificado del programa de Maestría en Entornos Virtuales de Aprendizaje, instalaciones adecuadas para el proceso de enseñanza-aprendizaje.

No se cuenta con más de un docente certificado y capacitado en el tema de Entornos Virtuales de Aprendizaje por lo que se necesitaría de presupuesto para becas para los docentes interesados en capacitarse y certificarse en el uso de los entornos virtuales.

Tabla 1. Desglose del presupuesto del proyecto educativo

Descripción	Costo
Laboratorio de computación (15 PCs)	¢25 000 000
Plataforma Moodle	Gratis
Capacitación Docente (C/persona)	¢2 500 000
Salario Docente (CM/persona)	¢1 200 000 x 5 meses = ¢6 000 000
Salario Técnico TI(CM/persona)	¢600 000 x 5 meses = ¢3 000 000
Total	¢36 500 000

10 Bibliografía

UTN (2017). *Maestría en Entornos Virtuales de Aprendizaje*. Recuperado de <http://www.utn.ac.cr/cfpte/maestr%C3%ADa-en-entornos-virtuales-0>

UTN (2017). *Modelo Educativo*. Recuperado de <http://www.utn.ac.cr/content/modelo-educativo>

Ingeniería Electrónica (2017). *Bachillerato en Electrónica*. Recuperado de <http://www.utn.ac.cr/content/ingenier%C3%ADa-electr%C3%B3nica>

DESARROLLO DEL PROYECTO

1. Nombre del curso virtual

Programación I

2. Selección y Justificación de las herramientas tecnológicas

Para la implementación del aula virtual del Curso: Programación I impartido por la Universidad Técnica Nacional, se utilizará la plataforma tecnológica Moodle, la cual es una herramienta tecnológica gratuita de código libre que puede ser utilizada, instalada, modificada entre otros por cualquier “WebMaster” que necesite instalarla en los servidores de una institución educativa o en un host en la nube para su acceso desde cualquier lugar.

Dicha plataforma es de las plataformas educativas más utilizadas a nivel mundial para impartir cursos virtuales, dado que al ser gratuita y configurable según las necesidades del usuario brinda un nivel de flexibilidad que atrae a todos los docentes virtuales dado que pueden editar y configurar el entorno virtual de aprendizaje según sus necesidades.

Por otro lado brinda al docente la posibilidad de escoger entre las diferentes opciones virtuales disponibles tales como foros, wikis, chats, recursos compartidos, trabajos colaborativos, Web 2.0, Web 3.0, compatibilidad de la plataforma a través de dispositivos móviles como teléfonos inteligentes, tablets, entre otros.

Por último otro de los factores de peso en la elección de la plataforma Moodle como herramienta para almacenar el entorno virtual de aprendizaje, es que el Centro de Formación Pedagógica y Tecnología Educativa de la Universidad Técnica Nacional es el ente regulador y administrador de la virtualidad dentro del campus universitario, donde se utiliza la plataforma virtual Moodle como herramienta tecnológica para todos los cursos de capacitación virtual impartidos por la casa de enseñanza.

Planteamiento de la estructura del entorno virtual

Foros: Tres foros evaluados, un foro de dudas generales y anuncios por todo el cuatrimestre. Un foro de ayuda por cada tarea programada.

Tareas: Tres tareas escritas de investigación que se deben subir al apartado de tareas

Tareas Programadas: Seis tareas programadas en todo el cuatrimestre que deben ser subidas al apartado de tareas programadas.

Correo Interno: Correo interno de la plataforma habilitado por todo el cuatrimestre para las consultas, mensajes enviados por el docente y estudiantes.

Aplicaciones y Web 2.0: Aplicaciones de Google para trabajos compartidos que pueden ser incrustados o compartidos al entorno virtual por medio de la Web, vídeos embebidos, presentaciones embebidas entre otros recursos virtuales que puedan ser compartidos por medio de la Web 2.0 al entorno virtual.

Guía Didáctica: Muestra todo un panorama del desarrollo del curso en todo el cuatrimestre, presentación del docente, cronograma de actividades, metodología, programa del curso.

Clases: Cuenta con siete clases distribuidas en todo el cuatrimestre de la siguiente forma

- 01- Estructuras de control selectivas
- 02- Estructuras de control repetitivas
- 03- Diagramas de Flujo
- 04- Vectores y Arreglos
- 05- Funciones y paso de parámetros por valor
- 06- Punteros y paso de parámetros por referencia
- 07- Estructuras y objetos

Archivos: En este apartado se encuentran los materiales del curso correspondiente a cada una de las clases y su respectiva unidad didáctica.

Sitios: URL de sitios complementarios de interés que el estudiante puede consultar para el complemento de los contenidos y ayuda en las tareas programadas.

3. Planificación de las clases

Para el desarrollo de los contenidos del Módulo #1 es necesario analizar las estructuras de control selectivas, estructuras de control repetitivas y los diagramas de flujo, lo que lleva a desarrollar tres clases que integren dichos contenidos en el proceso de aprendizaje.

Clase 1: Estructuras de control selectivas

- Objetivos de la clase:

Identificar y aplicar las estructuras de control selectivas al desarrollo de una aplicación de software que de solución a un problema.

- Contenidos de la clase:

- 1 Instrucción if (Definición e implementación)
- 2 Instrucción else if (Definición e implementación)
- 3 Instrucción else (Definición e implementación)
- 4 Instrucción case (Definición e implementación)

- Bibliografía de la clase:

Arias, Jonathan (2018). Estructuras de control y diagramas de flujo. Versión 1.0. Moodle UTN

- Recursos multimediales:

- Imagen que representa algunos lenguajes de programación: es una imagen donde muestra a un programador en contacto con una computadora para el desarrollo de un programa o aplicación de software. URL: <https://hipertextual.com/files/2015/09/Imagen-2-670x410.jpg>
- Video Tutorial 6 -Estructuras de control selectivas de c++ - sintaxis - if - switch - operador ternario: video publicado en YouTube que muestra el uso de las estructuras de control selectivas. URL: <https://www.youtube.com/watch?v=Xc6o4A7RfXc>

- Actividades: Implementación de un programa que de solución a una especificación.

Consigna de la Tarea Programada #1

Implementar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: El usuario debe ingresar un número natural (Conjunto de los Números Naturales) de cinco dígitos máximo en la entrada y obtener como resultado en la salida el equivalente en letras del número de entrada. El programa debe indicar al usuario cuando se ingrese un número fuera del rango solicitado.

Objetivo: Aplicar e implementar el uso de las estructuras de control selectivas en el desarrollo de una aplicación de software.

Evaluación: Se utilizará una rúbrica para la calificación de la Tarea Programada #1.

Plazos de entrega: Ocho días.

- Foro: Se abrirá un foro de debate en paralelo a la asignación para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas y retroalimentar con los foristas sus ideas y propuestas para la solución al problema.

- Consigna: Analizar como aplicar lo propuesto en el vídeo “Estructuras de Control Selectivas” al desarrollo de la aplicación de software solicitada en la Tarea Programada #1.
- Objetivo del Foro: Analizar y debatir sobre las estructuras de control selectivas aplicadas al desarrollo de una aplicación de software que de solución a un problema.
- Plazos de participación: Ocho días.
- Evaluación: Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Rúbrica de la Tarea Programada #1

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
Formato: Nitidez y presentación del código.	El código está ordenado e indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 3 puntos	El código No está ordenado pero sí indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 2 puntos	El código está ordenado pero No indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 1 punto	El código No esta ordenado y No está indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 0 puntos
Redacción y ortografía.	Respetar las reglas ortográficas y de redacción, en los comentarios del código.	Respetar las reglas ortográficas y de redacción, en los comentarios del código (1 a 2 errores).	Respetar las reglas ortográficas y de redacción, en los comentarios del código (3 a 4 errores).	Respetar las reglas ortográficas y de redacción, en los comentarios del código (5 o más errores).

	3 puntos	2 puntos	1 punto	0 puntos
Análisis del problema.	Analiza el problema modularmente implementando diagramas de flujo para la solución planteada.	Analiza el problema no modularmente pero sí implementa diagramas de flujo para la solución planteada.	Analiza el problema modularmente pero no utiliza diagramas de flujo para la solución planteada.	No analiza el problema modularmente ni implementa diagramas de flujo para la solución planteada.
	3 puntos	2 puntos	1 punto	0 puntos
Utilización de estructuras de control.	Implementa estructuras de control en la solución del problema.	Implementa estructuras de control en la solución del problema, pero presenta dificultades en el manejo de la información.	Implementa parcialmente estructuras de control en la solución del problema y presenta dificultades en el manejo de la información.	No utiliza estructuras de control en la solución del problema.
	3 puntos	2 puntos	1 punto	0 puntos
Implementación del algoritmo que	Implementación correcta del	Implementación parcial del	Implementación deficiente del	No implementa el algoritmo que

da solución al problema.	algoritmo que modela el comportamiento según lo solicitado. 3 puntos	algoritmo que modela el comportamiento según lo solicitado. 2 puntos	algoritmo que modela el comportamiento según lo solicitado. 1 punto	modela el comportamiento según lo solicitado. 0 puntos
Impresión de información.	Imprime en pantalla la información solicitada respetando el formato mostrado en el enunciado. 3 puntos	Imprime en pantalla la información solicitada con algunos errores respetando el formato mostrado en el enunciado. 2 puntos	Imprime en pantalla la información solicitada pero no respecta el formato mostrado en el enunciado. 1 punto	No implementa la impresión de información ni el formato solicitado. 0 puntos
Manejo de excepciones.	Implementa el manejo de excepciones en todo el programa y este no deja de funcionar tras ninguna excepción o error.	Implementa el manejo de excepciones en parcialmente el programa y este deja de funcionar en algunos casos de error.	Implementa el manejo de excepciones en menos de la mitad del programa y este deja de funcionar tras alguna excepción o error.	No implementa el manejo de excepciones en el programa.

	3 puntos	2 puntos	1 punto	0 puntos
Orden y claridad en el planteamiento.	La estructura e implementación del código fuente es clara y sigue el flujo diseñado en los diagramas de flujo.	La estructura e implementación del código fuente es confusa, pero sigue el flujo diseñado en los diagramas de flujo.	La estructura e implementación del código fuente es clara pero no sigue el flujo diseñado en los diagramas de flujo.	La estructura e implementación del código fuente no es clara y no sigue el flujo diseñado en los diagramas de flujo. 0 puntos
	3 puntos	2 puntos	1 punto	
Diagrama de flujo	Implementa adecuadamente y de forma completa el diagrama de flujo.	Implementa adecuadamente, pero de forma incompleta el diagrama de flujo.	Implementa con errores, pero de forma completa el diagrama de flujo.	No implementa el diagrama de flujo.
	3 puntos	2 puntos	1 puntos	0 puntos
Interfaz de usuario.	La aplicación es fácil de usar e intuitiva en su funcionamiento.	La aplicación es confusa de usar, pero se intuye el funcionamiento.	La aplicación es difícil de usar y poco intuitiva en su funcionamiento.	La aplicación es difícil de usar y no es intuitiva en su funcionamiento.

	3 puntos	2 puntos	1 punto	0 puntos
Total	30 puntos			0 puntos

Clase 2: Estructuras de control repetitivas

- Objetivos de la clase:

Identificar y aplicar las estructuras de control repetitivas al desarrollo de una aplicación de software que dé solución a un problema.

- Contenidos de la clase:

- Instrucción while (Definición e implementación)
- Instrucción do while (Definición e implementación)
- Instrucción for (Definición e implementación)

- Bibliografía de la clase:

Arias, Jonathan (2018). Estructuras de control y diagramas de flujo. Versión 1.0. Moodle UTN

- Recursos multimediales:

- Imagen que representa algunos lenguajes de programación: es una imagen donde muestra a un programador en contacto con una computadora para el desarrollo de un programa o aplicación de software. URL: <https://hipertextual.com/files/2015/09/Imagen-2-670x410.jpg>
- Video Estructuras de control repetitivas mientras repetir y Para: video publicado en YouTube que muestra el uso de las estructuras de control repetitivas. URL: https://www.youtube.com/watch?v=Fie8cZ_sWG8

- Actividades: Foro evaluado de análisis e intercambio de información

Consigna del Foro Evaluado #1

¿Qué opina sobre el uso de las estructuras de control en el ejemplo de la Figura 1? ¿Están bien aplicadas? ¿Es eficiente su implementación? ¿Utilizaría el ejemplo de la Figura 1 para aplicarlo a una solución real? ¿Lo cambiaría por completo o lo utilizaría? ¿Por qué? ¿Se debe implementar un diagrama de flujo? ¿Por qué?

```
1 /*Ejemplo #1 sobre estructuras de control */
2 #include <iostream>
3 using namespace std;
4 int main(void)
5 {
6     unsigned int nota;
7     cout << "Introduzca una calificacion numerica entre 0 y 10:";
8     cin >> nota;
9     cout << "La calificacion del alumno es" << endl;
10    if (nota == 10)
11    {
12        cout << "Matricula de Honor" << endl;
13    }
14    else if (nota >= 9)
15    {
16        cout << "Sobresaliente" << endl;
17    }
18    else if (nota >= 7)
19    {
20        cout << "Notable" << endl;
21    }
22    else if (nota >= 5)
23    {
24        cout << "Aprobado" << endl;
25    }
26    else
27    {
28        cout << "Suspenso" << endl;
29    }
30    system("pause");
31 }
```

Figura 1. Uso de las estructuras de control

Objetivo: Analizar y discutir el uso de las estructuras de control selectivas en el desarrollo de una aplicación de software.

Plazos de entrega: Ocho días.

Evaluación: Se utilizará una rúbrica para la calificación del foro evaluado.

- Foro: Se abrirá un foro de debate en paralelo a la asignación para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas y retroalimentar con los foristas sus ideas y propuestas para la solución al problema.

- **Consigna:** Analizar cómo aplicar lo propuesto en el vídeo “Estructuras de Control Repetitivas” al desarrollo de la aplicación de software solicitada en la Foro Evaluado #1.
- **Objetivo del Foro:** Analizar y debatir sobre las estructuras de control repetitivas aplicadas al desarrollo de una aplicación de software que de solución a un problema.
- **Plazos de participación:** Ocho días.
- **Evaluación:** Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Rúbrica del Foro Evaluado #1

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
Formato: Nitidez y presentación	El código está ordenado e indentado con 3	El código No está ordenado pero si indentado con 3	El código está ordenado pero No indentado con 3	El código No está ordenado y No esta indentado

del código.	espacios por cada inicio de nueva instrucción de código por línea. 3 puntos	espacios por cada inicio de nueva instrucción de código por línea. 2 puntos	espacios por cada inicio de nueva instrucción de código por línea. 1 punto	con 3 espacios por cada inicio de nueva instrucción de código por línea. 0 puntos
Redacción y ortografía.	Respetar las reglas ortográficas y de redacción, en los comentarios del foro. 3 puntos	Respetar las reglas ortográficas y de redacción, en los comentarios del foro (1 a 2 errores). 2 puntos	Respetar las reglas ortográficas y de redacción, en los comentarios del foro (3 a 4 errores). 1 punto	Respetar las reglas ortográficas y de redacción, en los comentarios del foro (5 o más errores). 0 puntos
Análisis del problema.	Analiza el problema modularmente implementando diagramas de flujo para la solución planteada. 3 puntos	Analiza el problema no modularmente, pero si implementa diagramas de flujo para la solución planteada. 2 puntos	Analiza el problema modularmente pero no utiliza diagramas de flujo para la solución planteada. 1 punto	No analiza el problema modularmente ni implementa diagramas de flujo para la solución planteada. 0 puntos

Participación efectiva en el foro	Realiza su aporte con base a lo solicitado y realiza por completo las réplicas a los compañeros.	Realiza su aporte con base a lo solicitado, pero no realiza por completo las réplicas a los compañeros.	No realiza por completo su aporte con base a lo solicitado, pero no realiza por completo las réplicas a los compañeros.	No realiza su aporte y no realiza por completo las réplicas a los compañeros.
	3 puntos	2 puntos	1 punto	0 puntos
Total	12 puntos			0 puntos

Clase 3: Diagramas de flujo

- Objetivos de la clase:

Interpretar y aplicar los diagramas de flujo al desarrollo de una aplicación de software que de solución a un problema.

- Contenidos de la clase:

- Definición
- Símbolos utilizados
- Diseño de un diagrama de flujo
- Software para su implementación

- Bibliografía de la clase:

Arias, Jonathan (2018). Estructuras de control y diagramas de flujo. Versión 1.0. Moodle UTN

- Recursos multimediales:

- Imagen que representa algunos lenguajes de programación: es una imagen donde muestra a un programador en contacto con una computadora para el desarrollo de un programa o aplicación de software. URL: <https://hipertextual.com/files/2015/09/Imagen-2-670x410.jpg>
- Video Diagramas de flujo en Raptor paso a paso: video publicado en YouTube que muestra el uso de la herramienta raptor para el diseño de un diagrama de flujo. URL: <https://www.youtube.com/watch?v=YW-Hx5NoJ0Q>

- Actividades: Implementación de un programa que dé solución a una especificación

Consigna de la Tarea Programada #2

Desarrollar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: Implementar una calculadora básica que realice las siguientes operaciones que se muestran y acceden a través de un menú de opciones como se muestra a continuación:

1. Suma
2. Resta
4. Multiplicación
5. División
6. Potencia 2^n
7. Salir

Objetivo: Aplicar e implementar el uso de las estructuras de control selectivas y repetitivas en el desarrollo de una aplicación de software utilizando diversas herramientas para el desarrollo de un diagrama de flujo.

Evaluación: Se utilizará una rúbrica para la calificación de la Tarea Programada #2.

Plazos de entrega: Ocho días.

- Foro: Se abrirá un foro de debate en paralelo a la asignación para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas y retroalimentar con los foristas sus ideas y propuestas para la solución al problema.

- **Consigna:** Analizar cómo aplicar lo propuesto en el vídeo “Diagramas de Flujo en Raptor” al desarrollo de la aplicación de software solicitada en la Tarea Programada #1.
- **Objetivo del Foro:** Analizar y debatir sobre los diagramas de flujo aplicados al desarrollo de una aplicación de software que dé solución a un problema.
- **Plazos de participación:** Ocho días.
- **Evaluación:** Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Rúbrica de la Tarea Programada #2

Criterio	Cumple a satisfacción lo indicado en la evaluación	Cumple medianamente lo indicado en la evaluación	Cumple en contenido y formato, pero los aportes no son significantes	No cumple o no presenta lo solicitado
Formato: Nitidez	El código está	El código No	El código está	El código No esta

y presentación del código.	ordenado e indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 3 puntos	está ordenado pero sí indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 2 puntos	ordenado pero No indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 1 punto	ordenado y No está indentado con 3 espacios por cada inicio de nueva instrucción de código por línea. 0 puntos
Redacción y ortografía.	Respetar las reglas ortográficas y de redacción, en los comentarios del código. 3 puntos	Respetar las reglas ortográficas y de redacción, en los comentarios del código (1 a 2 errores). 2 puntos	Respetar las reglas ortográficas y de redacción, en los comentarios del código (3 a 4 errores). 1 punto	Respetar las reglas ortográficas y de redacción, en los comentarios del código (5 o más errores). 0 puntos
Análisis del problema.	Analiza el problema modularmente implementando diagramas de flujo para la solución	Analiza el problema no modularmente pero sí implementa diagramas de flujo para la solución	Analiza el problema modularmente pero no utiliza diagramas de flujo para la solución	No analiza el problema modularmente ni implementa diagramas de flujo para la solución

	planteada. 3 puntos	solución planteada. 2 puntos	planteada. 1 punto	planteada. 0 puntos
Utilización de estructuras de control.	Implementa estructuras de control en la solución del problema. 3 puntos	Implementa estructuras de control en la solución del problema, pero presenta dificultades en el manejo de la información. 2 puntos	Implementa parcialmente estructuras de control en la solución del problema y presenta dificultades en el manejo de la información. 1 punto	No utiliza estructuras de control en la solución del problema. 0 puntos
Implementación del algoritmo que da solución al problema.	Implementación correcta del algoritmo que modela el comportamiento según lo solicitado. 3 puntos	Implementación parcial del algoritmo que modela el comportamiento según lo solicitado. 2 puntos	Implementación deficiente del algoritmo que modela el comportamiento según lo solicitado. 1 punto	No implementa el algoritmo que modela el comportamiento según lo solicitado. 0 puntos
Impresión de	Imprime en	Imprime en	Imprime en	No implementa la

información.	pantalla la información solicitada respectando el formato mostrado en el enunciado.	pantalla la información solicitada con algunos errores respectando el formato mostrado en el enunciado.	pantalla la información solicitada pero no respecta el formato mostrado en el enunciado.	impresión de información ni el formato solicitado.
	3 puntos	2 puntos	1 punto	0 puntos
Manejo de excepciones.	Implementa el manejo de excepciones en todo el programa y este no deja de funcionar tras ninguna excepción o error.	Implementa el manejo de excepciones en parcialmente el programa y este deja de funcionar en algunos casos de error.	Implementa el manejo de excepciones en menos de la mitad del programa y este deja de funcionar tras alguna excepción o error.	No implementa el manejo de excepciones en el programa.
	3 puntos	2 puntos	1 punto	0 puntos
Orden y claridad en el planteamiento.	La estructura e implementación del código fuente es clara y sigue el flujo diseñado en los	La estructura e implementación del código fuente es confusa, pero sigue el flujo	La estructura e implementación del código fuente es clara pero no sigue el flujo diseñado en los	La estructura e implementación del código fuente no es clara y no sigue el flujo diseñado en los

	diagramas de flujo.	diseñado en los diagramas de flujo.	diagramas de flujo.	diagramas de flujo.
	3 puntos	2 puntos	1 punto	0 puntos
Diagrama de flujo	Implementa adecuadamente y de forma completa el diagrama de flujo.	Implementa adecuadamente , pero de forma incompleta el diagrama de flujo.	Implementa con errores, pero de forma completa el diagrama de flujo.	No implementa el diagrama de flujo.
	3 puntos	2 puntos	1 puntos	0 puntos
Interfaz de usuario.	La aplicación es fácil de usar e intuitiva en su funcionamiento.	La aplicación es confusa de usar, pero se intuye el funcionamiento.	La aplicación es difícil de usar y poco intuitiva en su funcionamiento.	La aplicación es difícil de usar y no es intuitiva en su funcionamiento.
	3 puntos	2 puntos	1 punto	0 puntos
Total	30 puntos			0 puntos

4. Redacción de las clases

Clase 1: Estructuras de Control Parte I



Estimadas y estimados estudiantes,

Les damos la bienvenida a nuestra primera clase del curso Programación I, en la cual les propondremos una serie de conceptos relacionados con el paradigma de programación estructurada aplicada al lenguaje de programación C/C++. Es de gran importancia que interiorice los conceptos que se plantearán en el [Módulo #1: Estructuras de Control y Diagramas de Flujo](#), dado que se seguirán utilizando por el resto de su vida en el área de programación de dispositivos electrónicos, por lo que los invitamos a reflexionar sobre este tema y las aplicaciones en el mundo de la electrónica digital.

Para el desarrollo de la clase, debe leer la primera parte del [Módulo #1: Estructuras de Control y Diagramas de Flujo](#) para obtener los conocimientos previos básicos sobre el tema para poder desarrollar y aplicar los conceptos teóricos. Al finalizar la lectura tendrá las herramientas básicas para comprender que son las estructuras de control selectivas y como se aplican estas en la solución de un problema real de ingeniería aplicado.

Actividades

Cada participante de forma individual debe desarrollar un programa bajo un ambiente de desarrollo en el sistema operativo Linux que cumpla con lo solicitado. Para la revisión de la asignación el programa debe compilar sin errores ni warnings. En caso contrario, no será calificada la asignación.

Implementar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: El usuario debe ingresar un número natural (Conjunto de los Números Naturales) de cinco dígitos máximo en la entrada y obtener como resultado en la salida el equivalente en letras del número de entrada. El programa debe indicar al usuario cuando se ingrese un número fuera del rango solicitado.

El programa debe manejar todas las posibles excepciones que provoquen que este deje de funcionar por efecto del usuario. La única excepción permitida es el ingreso de letras por parte del usuario.

El objetivo de esta asignación es que pueda aplicar e implementar el uso de las estructuras de control selectivas en el desarrollo de una aplicación de software que dé solución al problema planteado.

La asignación debe ser subida al apartado de **Tarea Programada #1**, comprimida en un archivo RAR que internamente respete el siguiente formato:

Carpeta: <PrimerApellido>_<Nombre>_TareaProgramada1

Por ejemplo, la carpeta en mi caso resultaría: Arias_Jonathan_TareaProgramada1

Archivo comprimido: <PrimerApellido>_<Nombre>_TareaProgramada1.rar

Por ejemplo, el archivo en mi caso resultaría: Arias_Jonathan_TareaProgramada1.rar

Código fuente: <PrimerApellido>_<Nombre>_TareaProgramada1.c

Por ejemplo, el código fuente en mi caso resultaría: Arias_Jonathan_TareaProgramada1.c

Script de compilación: <PrimerApellido>_<Nombre>_Compilador.sh

Por ejemplo, el script en mi caso resultaría: Arias_Jonathan_Compilador.sh

Plazos de entrega: Cuenta con ocho días a partir de la fecha y hora de publicación.

Esta asignación tiene un valor de 3% de la nota final del curso y será evaluada con una rúbrica que puede acceder a través del siguiente enlace [Rubrica_Clases.pdf](#).

Por otro lado, cuenta con un foro de debate [Foro de Debate: TareaProgramada#1](#) que fue abierto en paralelo para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Saludos

[Jonathan](#)

[Foro de Debate: TareaProgramada#1](#)

Estimadas y estimados estudiantes,

El presente foro va dirigido a ustedes que actualmente están desarrollando la [Tarea Programada #1](#) con el objetivo que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Analice como aplicar lo propuesto en el vídeo “Tutorial 6 -Estructuras de control selectivas de c++ - sintaxis - if - switch - operador ternario”(Video incrustado en formato Web 2.0, URL: <https://www.youtube.com/watch?v=Xc6o4A7RfXc>) al desarrollo de la aplicación de software solicitada en la [Tarea Programada #1](#).

Debe realizar un aporte propio y tres réplicas al aporte propio de cada compañero(a).

Cuenta con ocho días a partir de la fecha y hora de publicación.

Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Saludos

Jonathan

Clase 2: Estructuras de Control Parte II



Estimadas y estimados estudiantes,

Iniciamos nuestra segunda clase sobre el tema de estructuras de control repetitivas que es una continuación de la primera clase, donde pondrá en práctica el tema de las iteraciones finitas y ciclos infinitos que son de gran importancia en el modelado del comportamiento de un proceso u actividad de la vida real por medio de una solución iterativa a través del software orientado a este tipo de procesos.

Para el desarrollo de la clase debe leer la segunda parte del **Módulo #1: Estructuras de Control y Diagramas de Flujo** para obtener los conocimientos sobre iteraciones finitas e infinitas para poder desarrollar y aplicar los conceptos teóricos. Una vez concluida la lectura de la segunda parte del módulo, podrá comprender que son las estructuras de control repetitivas y como se aplican estas en la solución de un problema real de ingeniería aplicado.

Actividades

El presente foro va dirigido a ustedes que actualmente están matriculados en el curso de Programación I, donde están iniciando su camino en el mundo de la programación estructurada, formando sus primeras bases de programación y dando sus primeros pasos en el mundo del software.

Este foro tiene como objetivo que desarrollen y apliquen las estructuras de control selectivas y repetitivas, al lenguaje de programación C/C++ como herramienta de programación utilizada en la solución de un problema de software.

Pretendemos que utilicen las estructuras de control en casos aplicados en la vida real sobre el uso de las estructuras de control para que interiorice y visualice la aplicación de los conceptos en el diario vivir del mundo de la programación.

Analice la siguiente consigna planteada

¿Qué opina sobre el uso de las estructuras de control en el ejemplo de la Figura 1? ¿Están bien aplicadas? ¿Es eficiente su implementación? ¿Utilizaría el ejemplo de la Figura 1 para aplicarlo a una solución real? ¿Lo cambiaría por completo o lo utilizaría? ¿Por qué? ¿Se debe implementar un diagrama de flujo? ¿Por qué?

```

1 /*Ejemplo #1 sobre estructuras de control */
2 #include <iostream>
3 using namespace std;
4 int main(void)
5 {
6     unsigned int nota;
7     cout << "Introduzca una calificacion numerica entre 0 y 10:";
8     cin >> nota;
9     cout << "La calificacion del alumno es" << endl;
10    if (nota == 10)
11    {
12        cout << "Matricula de Honor" << endl;
13    }
14    else if (nota >= 9)
15    {
16        cout << "Sobresaliente" << endl;
17    }
18    else if (nota >= 7)
19    {
20        cout << "Notable" << endl;
21    }
22    else if (nota >= 5)
23    {
24        cout << "Aprobado" << endl;
25    }
26    else
27    {
28        cout << "Suspenso" << endl;
29    }
30    system("pause");
31 }

```

Figura 1. Uso de las estructuras de control

Debe argumentar y demostrar porque Sí o porque No están bien aplicadas las estructuras de control en el ejemplo de la Figura 1 y, en caso de no estar bien aplicadas, proponer los cambios necesarios para mejorar el rendimiento en la ejecución del programa realizando una comparación con los tiempos de ejecución o tiempos de acceso a memoria según corresponda de ambos programas (Versión original vs Versión mejorada). Además, analizar porque Sí o porque No se debe implementar y utilizar un diagrama de flujo.

Plazos de entrega: Cuenta con ocho días a partir de la fecha y hora de publicación.

Esta asignación tiene un valor de 6% de la nota final del curso y será evaluada con una rúbrica que puede acceder a través del siguiente enlace [Rubrica_Clases.pdf](#).

Por otro lado, cuenta con un foro de debate **Foro de Debate: Foro_Evaluado#1** que fue abierto en paralelo para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Saludos

Jonathan

Foro de Debate: Foro_Evaluado#1

Estimadas y estimados estudiantes,

Este nuevo foro de debate que complementa al **Foro_Evaluado #1** tiene como objetivo que cada uno de ustedes pueda debatir sobre la temática a desarrollar, aclarar sus dudas, retroalimentar con sus compañeros ideas y propuestas para buscar una solución al tema planteado.

Analice como aplicar lo propuesto en el vídeo “Estructuras de control repetitivas mientras repetir y Para” (Video incrustado en formato Web 2.0, URL: https://www.youtube.com/watch?v=Fie8cZ_sWG8 al desarrollo de la aplicación de software solicitada en el **Foro_Evaluado #1**.

Debe realizar un aporte propio y tres réplicas al aporte propio de cada compañero(a).

Cuenta con ocho días a partir de la fecha y hora de publicación.

Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Saludos

Jonathan

Clase 3: Diagramas de Flujo



Estimadas y estimados estudiantes,

Llegamos al final del primer módulo (tercera parte), donde en esta tercera clase tiene que analizar, aplicar los contenidos del diseño y desarrollo de diagramas de flujo en la implementación de una solución de software, sus ventajas, beneficios, eficiencia que ofrece al programador invertir tiempo antes de iniciar con el desarrollo de la aplicación el diseño del diagrama de flujo.

Para esta tercera clase en curso debe leer la tercera parte del [Módulo #1: Estructuras de Control y Diagramas de Flujo](#) para interiorizar los conceptos sobre el tema para poder desarrollar y aplicarlos en el desarrollo de un programa. Al finalizar la lectura se espera pueda aplicar correctamente los diagramas de flujo, las estructuras de control selectivas, repetitivas y como se aplica todo esto en la solución de un problema de software.

Actividades

Cada participante de forma individual debe desarrollar un programa bajo un ambiente de desarrollo en el sistema operativo Linux que cumpla con lo solicitado. Para la revisión de la asignación el programa debe compilar sin errores ni warnings. En caso contrario no será calificada la asignación.

Desarrollar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: Implementar una calculadora básica que realice las siguientes operaciones que se muestran y acceden a través de un menú de opciones como se muestra a continuación:

1. Suma
2. Resta
4. Multiplicación
5. División
6. Potencia 2ⁿ
7. Salir

El usuario debe poder seleccionar cada una de las seis primeras opciones, solicitar el ingreso de los dos parámetros de entrada (números enteros), visualizar el resultado de la operación y volver al menú principal cada vez que se termine de ejecutar la operación aritmética. Cuando se seleccione la opción Salir, el programa finaliza y se despide con un mensaje antes de finalizar. El programa debe indicar cuando se ingresa una opción fuera de rango en el menú y volver a solicitar la opción de entrada al usuario. Debe implementar el diagrama de flujo de la solución en al menos dos herramientas de software distintas.

El programa debe manejar todas las posibles excepciones que provoquen que este deje de funcionar por efecto del usuario. La única excepción permitida es el ingreso de letras por parte del usuario.

El objetivo de esta asignación es que pueda aplicar e implementar el uso de las estructuras

de control selectivas, repetitivas y diagramas de flujo en el desarrollo de una aplicación de software que dé solución al problema planteado.

La asignación debe ser subida al apartado de [Tarea Programada #2](#), comprimida en un archivo RAR que internamente respete el siguiente formato:

Carpeta: <PrimerApellido>_<Nombre>_TareaProgramada2

Por ejemplo, la carpeta en mi caso resultaría: Arias_Jonathan_TareaProgramada2

Archivo comprimido: <PrimerApellido>_<Nombre>_TareaProgramada2.rar

Por ejemplo, el archivo en mi caso resultaría: Arias_Jonathan_TareaProgramada2.rar

Código fuente: <PrimerApellido>_<Nombre>_TareaProgramada2.c

Por ejemplo, el código fuente en mi caso resultaría: Arias_Jonathan_TareaProgramada2.c

Script de compilación: <PrimerApellido>_<Nombre>_Compilador.sh

Por ejemplo, el script en mi caso resultaría: Arias_Jonathan_Compilador.sh

Plazos de entrega: Cuenta con ocho días a partir de la fecha y hora de publicación.

Esta asignación tiene un valor de 3% de la nota final del curso y será evaluada con una rúbrica que puede acceder a través del siguiente enlace [Rubrica_Clases.pdf](#).

Por otro lado, cuenta con un foro de debate [Foro de Debate: TareaProgramada#2](#) que fue abierto en paralelo para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Saludos

Jonathan

Foro de Debate: TareaProgramada#2

Estimadas y estimados estudiantes,

Este foro de debate final sobre el **Módulo #1: Estructuras de Control y Diagramas de Flujo** complementa el desarrollando la **Tarea Programada #2** con el objetivo que cada uno pueda debatir sobre el tema en desarrollo, aclarar dudas, retroalimentar ideas entre los miembros del curso para llegar a un consenso entre la comunidad.

Analice como aplicar lo propuesto en el vídeo “Diagramas de flujo en Raptor paso a paso” (Video incrustado en formato Web 2.0, URL: <https://www.youtube.com/watch?v=YW-Hx5NoJ0Q>) al desarrollo de la aplicación de software solicitada en la **Tarea Programada #2**.

Debe realizar un aporte propio y tres réplicas al aporte propio de cada compañero(a).

Cuenta con ocho días a partir de la fecha y hora de publicación.

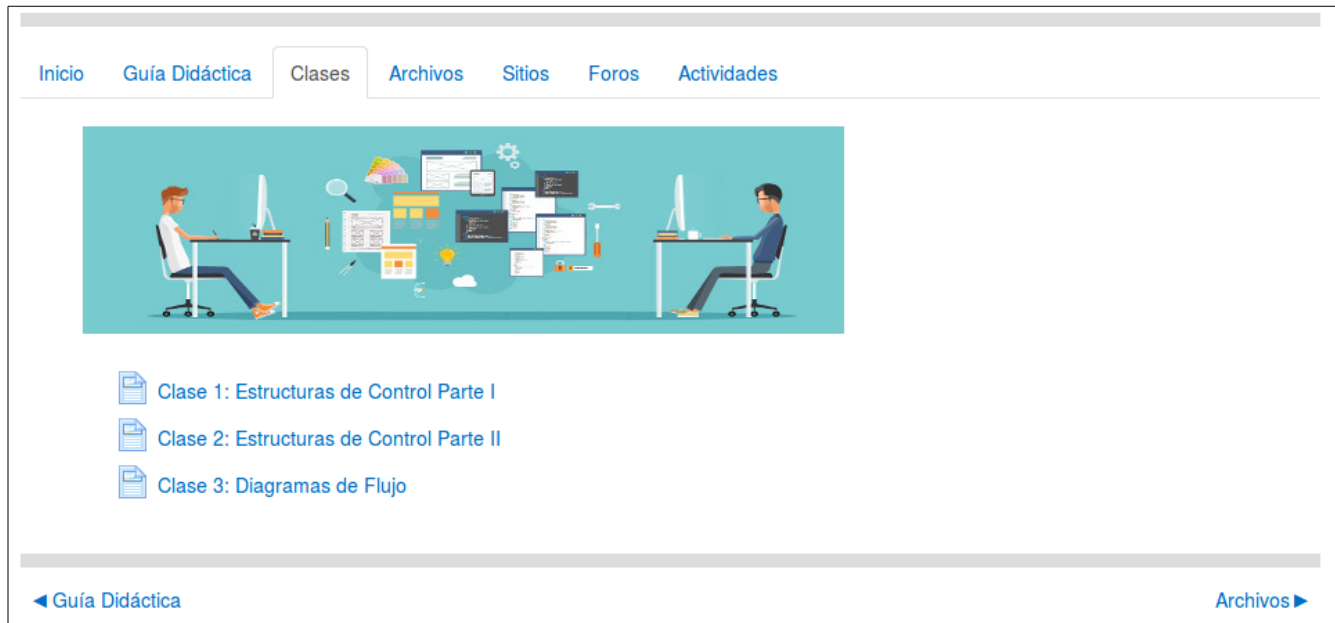
Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Saludos

Jonathan

5. Captura de pantalla de las clases

Inicio de Clases




The screenshot shows a website interface with a navigation menu at the top containing links for 'Inicio', 'Guía Didáctica', 'Clases', 'Archivos', 'Sitios', 'Foros', and 'Actividades'. The 'Clases' link is highlighted. Below the menu is a large illustration of two people working at computers with various icons representing programming and design. Underneath the illustration, there is a list of three class items, each with a document icon: 'Clase 1: Estructuras de Control Parte I', 'Clase 2: Estructuras de Control Parte II', and 'Clase 3: Diagramas de Flujo'. At the bottom of the page, there are navigation arrows for 'Guía Didáctica' and 'Archivos'.

Clase #1



Clase 1: Estructuras de Control Parte I



Estimadas y estimados estudiantes,

Les damos la bienvenida a nuestra primera clase del curso Programación I, en la cual les propondremos una serie de conceptos relacionados con el paradigma de programación estructurada aplicada al lenguaje de programación C/C++. Es de gran importancia que interiorice los conceptos que se plantearán en el [Módulo #1: Estructuras de Control y Diagramas de Flujo](#), dado que se seguirán utilizando por el resto de su vida en el área de programación de dispositivos electrónicos, por lo que los invitamos a reflexionar sobre este tema y las aplicaciones en el mundo de la electrónica digital.

Para el desarrollo de la clase, debe leer la primera parte del [Módulo #1: Estructuras de Control y Diagramas de Flujo](#) para obtener los conocimientos previos básicos sobre el tema para poder desarrollar y aplicar los conceptos teóricos. Al finalizar la lectura tendrá las herramientas básicas para comprender que son las [estructuras de control selectivas](#) y como se aplican estas en la solución de un problema real de ingeniería aplicado.

Actividades

Cada participante de forma individual debe desarrollar un programa bajo un ambiente de desarrollo en el sistema operativo Linux que cumpla con lo solicitado. Para la revisión de la asignación el programa debe compilar sin errores ni warnings. En caso contrario, no será calificada la asignación.

Actividades

Cada participante de forma individual debe desarrollar un programa bajo un ambiente de desarrollo en el sistema operativo Linux que cumpla con lo solicitado. Para la revisión de la asignación el programa debe compilar sin errores ni warnings. En caso contrario, no será calificada la asignación.

Implementar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: El usuario debe ingresar un número natural (Conjunto de los Números Naturales) de cinco dígitos máximo en la entrada y obtener como resultado en la salida el equivalente en letras del número de entrada. El programa debe indicar al usuario cuando se ingrese un número fuera del rango solicitado.

El programa debe manejar todas las posibles excepciones que provoquen que este deje de funcionar por efecto del usuario. La única excepción permitida es el ingreso de letras por parte del usuario.

El objetivo de esta asignación es que pueda aplicar e implementar el uso de las [estructuras de control selectivas](#) en el desarrollo de una aplicación de software que dé solución al problema planteado.

La asignación debe ser subida al apartado de [Tarea Programada #1](#), comprimida en un archivo RAR que internamente respete el siguiente formato:

Carpeta: <PrimerApellido>_<Nombre>_TareaProgramada1
Por ejemplo, la carpeta en mi caso resultaría: Arias_Jonathan_TareaProgramada1

Archivo comprimido: <PrimerApellido>_<Nombre>_TareaProgramada1.rar
Por ejemplo, el archivo en mi caso resultaría: Arias_Jonathan_TareaProgramada1.rar

Código fuente: <PrimerApellido>_<Nombre>_TareaProgramada1.c
Por ejemplo, el código fuente en mi caso resultaría: Arias_Jonathan_TareaProgramada1.c

Script de compilación: <PrimerApellido>_<Nombre>_Compilador.sh
Por ejemplo, el script en mi caso resultaría: Arias_Jonathan_Compilador.sh

Plazos de entrega: Cuenta con ocho días a partir de la fecha y hora de publicación.

Esta asignación tiene un valor de 3% de la nota final del curso y será evaluada con una rúbrica que puede acceder a través del siguiente enlace [Rubrica_Clases.pdf](#).

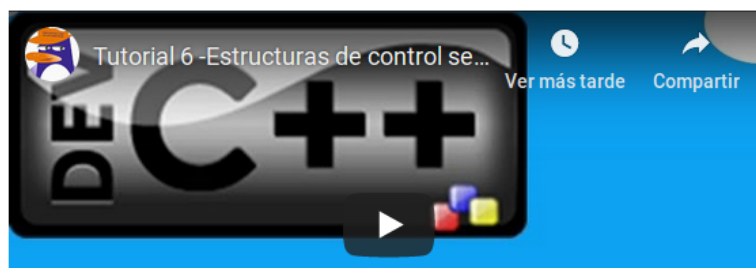
Por otro lado, cuenta con un foro de debate [Foro de Debate: TareaProgramada#1](#) que fue abierto en paralelo para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Foro de Debate: TareaProgramada#1

Estimadas y estimados estudiantes,

El presente foro va dirigido a ustedes que actualmente están desarrollando la [Tarea Programada #1](#) con el objetivo que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Analice como aplicar lo propuesto en el vídeo "Tutorial 6 -Estructuras de control selectivas de c++ - sintaxis - if - switch - operador ternario" al desarrollo de la aplicación de software solicitada en la [Tarea Programada #1](#).



Analice como aplicar lo propuesto en el vídeo "Tutorial 6 -Estructuras de control selectivas de c++ - sintaxis - if - switch - operador ternario" al desarrollo de la aplicación de software solicitada en la [Tarea Programada #1](#).



Debe realizar un aporte propio y tres réplicas al aporte propio de cada compañero(a).

Cuenta con ocho días a partir de la fecha y hora de publicación.

Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Saludos
Jonathan

Clase #2

Clase 2: Estructuras de Control Parte II



Estimadas y estimados estudiantes,

Iniciamos nuestra segunda clase sobre el tema de **estructuras de control** repetitivas que es una continuación de la primera clase, donde pondrá en práctica el tema de las iteraciones finitas y ciclos infinitos que son de gran importancia en el modelado del comportamiento de un proceso u actividad de la vida real por medio de una solución iterativa a través del software orientado a este tipo de procesos.

Para el desarrollo de la clase debe leer la segunda parte del **Módulo #1: Estructuras de Control y Diagramas de Flujo** para obtener los conocimientos sobre iteraciones finitas e infinitas para poder desarrollar y aplicar los conceptos teóricos. Una vez concluida la lectura de la segunda parte del módulo, podrá comprender que son las **estructuras de control** repetitivas y como se aplican estas en la solución de un problema real de ingeniería aplicado.

Actividades

Foro Evaluado #1

El presente foro va dirigido a ustedes que actualmente están matriculados en el curso de Programación I, donde están iniciando su camino en el mundo de la programación estructurada, formando sus primeras bases de programación y dando sus primeros pasos en el mundo del software.

Este foro tiene como objetivo que desarrollen y apliquen las **estructuras de control selectivas y repetitivas**, al lenguaje de programación C/C++ como

Actividades

Foro Evaluado #1

El presente foro va dirigido a ustedes que actualmente están matriculados en el curso de Programación I, donde están iniciando su camino en el mundo de la programación estructurada, formando sus primeras bases de programación y dando sus primeros pasos en el mundo del software.

Este foro tiene como objetivo que desarrollen y apliquen las **estructuras de control selectivas y repetitivas**, al lenguaje de programación C/C++ como herramienta de programación utilizada en la solución de un problema de software.

Pretendemos que utilicen las **estructuras de control** en casos aplicados en la vida real sobre el uso de las **estructuras de control** para que interiorice y visualice la aplicación de los conceptos en el diario vivir del mundo de la programación.

Analice la siguiente consigna planteada

¿Qué opina sobre el uso de las **estructuras de control** en el ejemplo de la Figura 1? ¿Están bien aplicadas? ¿Es eficiente su implementación? ¿Utilizaría el ejemplo de la Figura 1 para aplicarlo a una solución real? ¿Lo cambiaría por completo o lo utilizaría? ¿Por qué? ¿Se debe implementar un **diagrama de flujo**? ¿Por qué?

```
1 /*Ejemplo #1 sobre estructuras de control */
2 #include <iostream>
3 using namespace std;
4 int main(void)
5 {
6     unsigned int nota;
7     cout << "Introduzca una calificacion numerica entre 0 y 10:";
8     cin >> nota;
9     cout << "La calificacion del alumno es" << endl;
10    if (nota == 10)
11    {
12        cout << "Matricula de Honor" << endl;
13    }
14    else if (nota >= 9)
15    {
16        cout << "Sobresaliente" << endl;
17    }
```

```

14     else if (nota >= 9)
15     {
16         cout << "Sobresaliente" << endl;
17     }
18     else if (nota >= 7)
19     {
20         cout << "Notable" << endl;
21     }
22     else if (nota >= 5)
23     {
24         cout << "Aprobado" << endl;
25     }
26     else
27     {
28         cout << "Suspenso" << endl;
29     }
30     system("pause");
31 }

```

Figura 1. Uso de las estructuras de control

Debe argumentar y demostrar porque Sí o porque No están bien aplicadas las **estructuras de control** en el ejemplo de la Figura 1 y, en caso de no estar bien aplicadas, proponer los cambios necesarios para mejorar el rendimiento en la ejecución del programa realizando una comparación con los tiempos de ejecución o tiempos de acceso a memoria según corresponda de ambos programas (Versión original vs Versión mejorada). Además, analizar porque Sí o porque No se debe implementar y utilizar un **diagrama de flujo**.

Plazos de entrega: Cuenta con ocho días a partir de la fecha y hora de publicación.

Esta asignación tiene un valor de 6% de la nota final del curso y será evaluada con una rúbrica que puede acceder a través del siguiente enlace [Rubrica_Clases.pdf](#).

Por otro lado, **cuenta con un foro de debate** [Foro de Debate: Foro Evaluado#1](#) que fue abierto en paralelo para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Saludos
Jonathan

Clase #3

Clase 3: Diagramas de Flujo



Estimadas y estimados estudiantes,

Llegamos al final del primer módulo (tercera parte), donde en esta tercera clase tiene que analizar, aplicar los contenidos del diseño y desarrollo de diagramas de flujo en la implementación de una solución de software, sus ventajas, beneficios, eficiencia que ofrece al programador invertir tiempo antes de iniciar con el desarrollo de la aplicación el diseño del [diagrama de flujo](#).

Para esta tercera clase en curso debe leer la tercera parte del [Módulo #1: Estructuras de Control y Diagramas de Flujo](#) para interiorizar los conceptos sobre el tema para poder desarrollar y aplicarlos en el desarrollo de un programa. Al finalizar la lectura se espera pueda aplicar correctamente los diagramas de flujo, las [estructuras de control selectivas](#), repetitivas y como se aplica todo esto en la solución de un problema de software.

Actividades

Cada participante de forma individual debe desarrollar un programa bajo un ambiente de desarrollo en el sistema operativo Linux que cumpla con lo solicitado. Para la revisión de la asignación el programa debe compilar sin errores ni warnings. En caso contrario no será calificada la asignación.

Desarrollar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: Implementar una calculadora básica que realice las siguientes operaciones que se muestran y acceden a través de un menú de opciones como se muestra a continuación:

1. Suma

Actividades

Cada participante de forma individual debe desarrollar un programa bajo un ambiente de desarrollo en el sistema operativo Linux que cumpla con lo solicitado. Para la revisión de la asignación el programa debe compilar sin errores ni warnings. En caso contrario no será calificada la asignación.

Desarrollar un programa en el lenguaje de programación C/C++ que cumpla con las siguientes especificaciones: Implementar una calculadora básica que realice las siguientes operaciones que se muestran y acceden a través de un menú de opciones como se muestra a continuación:

1. Suma
2. Resta
4. Multiplicación
5. División
6. Potencia 2^n
7. Salir

El usuario debe poder seleccionar cada una de las seis primeras opciones, solicitar el ingreso de los dos parámetros de entrada (números enteros), visualizar el resultado de la operación y volver al menú principal cada vez que se termine de ejecutar la operación aritmética. Cuando se seleccione la opción Salir, el programa finaliza y se despide con un mensaje antes de finalizar. El programa debe indicar cuando se ingresa una opción fuera de rango en el menú y volver a solicitar la opción de entrada al usuario. Debe implementar el [diagrama de flujo](#) de la solución en al menos dos herramientas de software distintas.

El programa debe manejar todas las posibles excepciones que provoquen que este deje de funcionar por efecto del usuario. La única excepción permitida es el ingreso de letras por parte del usuario.

El objetivo de esta asignación es que pueda aplicar e implementar el uso de las [estructuras de control selectivas](#), repetitivas y diagramas de flujo en el desarrollo de una aplicación de software que dé solución al problema planteado.

La asignación debe ser subida al apartado de [Tarea Programada #2](#), comprimida en un archivo RAR que internamente respete el siguiente formato:

La asignación debe ser subida al apartado de **Tarea Programada #2**, comprimida en un archivo RAR que internamente respete el siguiente formato:

Carpeta: <PrimerApellido>_<Nombre>_TareaProgramada2
Por ejemplo, la carpeta en mi caso resultaría: Arias_Jonathan_TareaProgramada2

Archivo comprimido: <PrimerApellido>_<Nombre>_TareaProgramada2.rar
Por ejemplo, el archivo en mi caso resultaría: Arias_Jonathan_TareaProgramada2.rar

Código fuente: <PrimerApellido>_<Nombre>_TareaProgramada2.c
Por ejemplo, el código fuente en mi caso resultaría: Arias_Jonathan_TareaProgramada2.c

Script de compilación: <PrimerApellido>_<Nombre>_Compilador.sh
Por ejemplo, el script en mi caso resultaría: Arias_Jonathan_Compilador.sh

Plazos de entrega: Cuenta con ocho días a partir de la fecha y hora de publicación.

Esta asignación tiene un valor de 3% de la nota final del curso y será evaluada con una rúbrica que puede acceder a través del siguiente enlace [Rubrica_Clases.pdf](#).

Por otro lado, cuenta con un foro de debate [Foro de Debate: TareaProgramada#2](#) que fue abierto en paralelo para que cada estudiante pueda debatir sobre la temática en curso, aclarar dudas, retroalimentar con sus compañeros ideas y propuestas para la solución al problema.

Foro de Debate: TareaProgramada#2

Estimadas y estimados estudiantes,

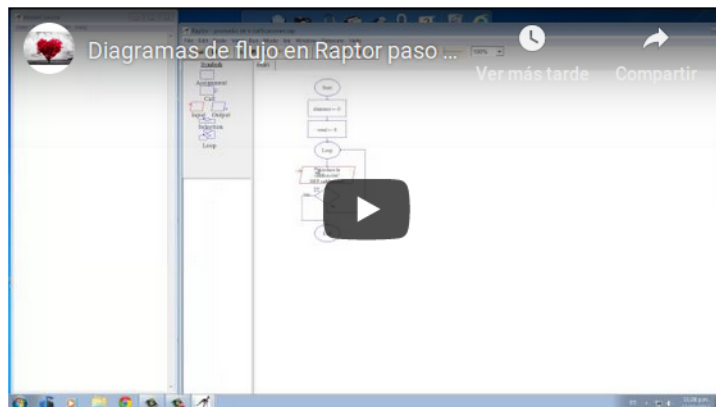
Este foro de debate final sobre el [Módulo #1: Estructuras de Control y Diagramas de Flujo](#) complementa el desarrollando la [Tarea_Programada #2](#) con el objetivo que cada uno pueda debatir sobre el tema en desarrollo, aclarar dudas, retroalimentar ideas entre los miembros del curso para

Foro de Debate: TareaProgramada#2

Estimadas y estimados estudiantes,

Este foro de debate final sobre el [Módulo #1: Estructuras de Control y Diagramas de Flujo](#) complementa el desarrollando la [Tarea_Programada #2](#) con el objetivo que cada uno pueda debatir sobre el tema en desarrollo, aclarar dudas, retroalimentar ideas entre los miembros del curso para llegar a un consenso entre la comunidad.

Analice como aplicar lo propuesto en el vídeo "Diagramas de flujo en Raptor paso a paso" al desarrollo de la aplicación de software solicitada en la [Tarea_Programada #2](#).



Debe realizar un aporte propio y tres réplicas al aporte propio de cada compañero(a).

Analice como aplicar lo propuesto en el vídeo "Diagramas de flujo en Raptor paso a paso" al desarrollo de la aplicación de software solicitada en la [Tarea_Programada #2](#).



Debe realizar un aporte propio y tres réplicas al aporte propio de cada compañero(a).

Cuenta con ocho días a partir de la fecha y hora de publicación.

Este foro es formativo y de participación obligatoria como requisito para dar por concluida la evaluación.

Saludos
Jonathan

DOCUMENTOS ELABORADOS

GUÍA DIDÁCTICA

1. Fundamentación de la materia¹

Este curso pretende que ustedes apliquen las nociones y los elementos básicos de un lenguaje de programación, orientado principalmente a la utilización del lenguaje de programación C++; mediante el diseño de algoritmos y la utilización de estructuras de decisión y control.

Las principales temáticas a desarrollar están enfocadas en los aspectos básicos de la programación, utilizando diferentes herramientas que permitirán conocer y familiarizarse con la aplicación, para posteriormente poder desarrollar soluciones diversas. La programación es un elemento fundamental en la educación de un Ingeniero en Electrónica, debido a que muchos dispositivos y elementos electrónicos digitales se programan con algún tipo de lenguaje de programación.¹

2. Objetivos Generales y Específicos¹

Objetivo General

- Aplicar los conocimientos básicos de programación en computadoras, mediante la realización de un proyecto programado, para el desarrollo de aplicaciones electrónicas avanzadas.

Objetivos Específicos

- Definir algoritmos, diagramas de flujo y sistemas numéricos, mediante la ejecución de ejercicios prácticos, para ordenar el proceso de programación en una computadora.

¹ Programa del Curso Programación I, Ingeniería Electrónica, UTN

- Distinguir los diferentes tipos de datos, variables y funciones en un programa específico, mediante la realización de prácticas en la computadora, para facilitar la implementación de programas funcionales.
- Diferenciar las estructuras de decisión y control, mediante la resolución de ejercicios de programación, para mejorar la optimización y funcionalidad de un programa.
- Aplicar las herramientas de programación en C++, mediante la realización de un proyecto práctico, para establecer la comunicación con un puerto de la computadora.

3. Contenidos

Módulo #1: Estructuras de Control y Diagramas de Flujo

- Estructuras de control selectivas: if, else if, else, case
- Estructuras de control repetitivas: while, do while, for
- Diagramas de Flujo: símbolos y software para su implementación

Módulo #2: Vectores y Arreglos

- Definición, implementación, recorrido, lectura, escritura de un vector
- Definición, implementación, recorrido, lectura, escritura de un arreglo

Módulo #3: Funciones y paso de parámetros por valor

- Definición, implementación y retorno de valores en una función
- Parámetros por valor como entradas de una función

Módulo #4: Punteros y paso de parámetros por referencia

- Definición, implementación, direccionamiento y aplicación de un puntero
- Parámetros por referencia como entradas y salidas de una función

Módulo #5: Estructuras de Datos y como aplicarlas

- Definición, implementación y acceso a una estructura(struct)
- Arreglos de estructuras y aplicaciones

4. Metodología de Trabajo¹

Se utilizará una metodología participativa por medio de un Entorno Virtual de Aprendizaje (EVA), con interacción docente-estudiante a través de la realización de foros, trabajos colaborativos y tareas programadas que serán subidas por el estudiante a la plataforma virtual para la revisión correspondiente.

Se habilitará un foro de consultas generales que permanecerá abierto por todo el curso para que el estudiante pueda comunicarse con los compañeros del curso y el docente en temas generales referentes al curso, notas, ente otros.

Se habilitará un foro de dudas por cada clase para que el estudiante pueda aclarar sus dudas e interactuar con el docente y los compañeros del curso sobre la clase en desarrollo.

Se trabajará con cinco módulos que se irán desarrollando y habilitando en transcurso del curso. Cada módulo tiene asignado diferentes clases y actividades las cuales deberá desarrollar en los tiempos establecidos.

En paralelo se trabajará con seis Tareas Programadas y tres foros evaluados que complementarán los contenidos desarrollados en cada una de las clases, junto con un Proyecto Final Programado que integrará todos los contenidos cubiertos en el curso.

Se realizarán dos pruebas, la primera prueba en semana siete y la segunda prueba en semana catorce. Ambas pruebas consisten en desarrollar una aplicación de software y subirla al apartado de "Pruebas" en el entorno virtual.

En paralelo a las clases, se habilitará material complementario de lectura para ayudar a reforzar los contenidos vistos en cada clase, la lectura no es obligatoria.

La entrega de las notas y recomendaciones a las "Tareas Programadas" será en los próximos ocho días siguientes a la fecha de entrega de la tarea.

5. Evaluación de los aprendizajes

Rubro	Porcentaje
Primer Examen	20%
Segundo Examen	20%
Tareas Programadas	20%
Foros	20%
Proyecto Final	20%
Total	100%

La nota mínima para la aprobación del curso es de 70 y no existe prueba de sustitución, dado que, al ser un curso de naturaleza práctica, no aplica el examen de sustitución según los lineamientos de evaluación de la UTN.

Se realizarán dos pruebas con un valor de 20% cada una, distribuidas en las semanas siete y catorce del cuatrimestre, para un total de un 40% de la nota del curso.

Se desarrollarán seis tareas programadas, las cinco primeras con un valor porcentual de 3% cada una y la sexta tarea con un valor de 5%, para un total de un 20% de la nota del curso.

Se desarrollarán tres foros evaluados, los primeros dos con un valor de 6% cada uno y el tercer foro con un valor de 8%, para un total de un 20% de la nota del curso.

Se desarrollará un Proyecto Final que integra todos los contenidos del curso, para aplicar todos los temas en la implementación de una aplicación de software que de solución a un problema de ingeniería, para un total de un 20% de la nota del curso.

Se tomarán en cuenta los siguientes Criterios de Evaluación

- Puntualidad: realiza la entrega de las asignaciones en la fecha establecida.
- Organización: se utilizan los recursos de forma adecuada y refleja un orden en la realización de la actividad.
- Uso de arreglos: maneja adecuadamente el uso de los arreglos en la solución de un problema.
- Uso de estructuras de control: maneja adecuadamente el uso de las estructuras de control en la solución de un problema.
- Uso de punteros: maneja adecuadamente el uso de los punteros como parámetros por referencia en una función para optimizar el consumo de memoria.
- Comentarios: realiza comentarios en todo el código que reflejen una lectura y comprensión clara para el lector.
- Indentación: aplica adecuadamente tres espacios de indentación por cada nueva línea de instrucción del código.

6. Cronograma de Trabajo

Semana	Actividad
1	Se habilita el Módulo #1, debe proceder con la lectura del módulo y evacuar sus dudas en el foro "Foro Módulo #1" que fue abierto en paralelo con la clase.
2	Se habilita la Clase #1-M1 donde debe realizar la Tarea Programada #1
3	Se habilita la Clase #2-M1 donde debe realizar el Foro Evaluado #1
4	Se habilita la Clase #3-M1 donde debe realizar la Tarea Programada #2
5	Se habilita el Módulo #2, debe proceder con la lectura del módulo y evacuar sus dudas en el foro "Foro Módulo #2" que fue abierto en paralelo con la clase. Se habilita la Clase #1-M2 donde debe realizar la Tarea Programada #3.
6	Se habilita el Módulo #3, debe proceder con la lectura del módulo y evacuar sus dudas en el foro "Foro Módulo #3" que fue abierto en paralelo con la clase.
7	Primer Examen
8	Se habilita la Clase #1-M3 donde debe realizar el Foro Evaluado #2
9	Se habilita el Módulo #4, debe proceder con la lectura del módulo y evacuar sus dudas en el foro "Foro Módulo #4" que fue abierto en paralelo con la clase. Se habilita la Clase #1-M4 donde debe realizar la Tarea Programada #4.
10	Se habilita la Clase #2-M4 donde debe realizar la Tarea Programada #5
11	Se habilita el Módulo #5, debe proceder con la lectura del módulo y evacuar sus dudas en el foro "Foro Módulo #5" que fue abierto en paralelo con la clase. Se habilita la Clase #1-M5 donde debe realizar el Foro

	Evaluado #3
12	Se habilita la Clase #2-M5 donde debe realizar la Tarea Programada #6
13	Se habilita la semana completa para trabajar en el proyecto, aclarar dudas, revisar avances del proyecto, recomendaciones del tutor, entre otros por cada estudiante.
14	Segundo Examen
15	Entrega Proyecto Final

7. Presentación del tutor

Hola estimados estudiantes, mi nombre es Jonathan Arias Román y voy a ser su profesor tutor en esta nueva aventura que inician en el mundo de la programación estructurada.



Como parte de mi experiencia profesional he trabajado en el área de Ingeniería Electrónica en proyectos de mantenimiento de máquinas eléctricas, automatización industrial en el sector privado, diseño de sistemas digitales y aplicaciones de software en diferentes empresas transnacionales en Costa Rica. Además, he trabajado en el sector educación para el Ministerio de Educación Pública (MEP) en la Especialidad Técnica Electrotecnia y como profesor en la Universidad Técnica Nacional (UTN) impartiendo mayoritariamente los cursos de programación y sistemas digitales.

Con respecto a mis logros académicos, soy graduado de Licenciatura en Ingeniería Electrónica del Instituto Tecnológico de Costa Rica, Licenciatura en Docencia de la Universidad San Marcos, Maestría en Administración Educativa de la Universidad Americana, Maestría en Entornos Virtuales de Aprendizaje de la Universidad Técnica Nacional, Diplomado en Informática de la Universidad Estatal a Distancia, actualmente cursando el Bachillerato en Ingeniería Informática de la Universidad Estatal a Distancia.

Espero que juntos disfrutemos esta travesía en el mundo de la programación para que puedan llegar a aplicar los conceptos del curso en el diseño y desarrollo de una aplicación de software que dé solución a un problema de ingeniería electrónica.

Cualquier duda o inquietud no duden en contactarme por medio de la plataforma virtual por medio de mensaje privado o a través de las diferentes opciones de foros disponibles.

Saludos

Jonathan





UTN

Universidad
Técnica Nacional

Escuela de Ingeniería Electrónica

Curso: Programación I

Código: IEL-721

Profesor: MSc.Ing. Jonathan Arias Román

I Cuatrimestre 2019

Tabla de Contenidos

Estructuras de Control: Parte I.....	3
Iniciamos nuestro camino.....	3
Instrucción de selección if.....	5
Instrucción de selección if...else.....	7
Instrucción de repetición while.....	8
Estructuras de Control: Parte II.....	10
Instrucción de repetición while.....	10
Instrucción de repetición for.....	12
Instrucción de repetición do...while.....	15
Diagramas de Flujo.....	17
Ventajas de utilizar diagramas de flujo.....	17
Símbolos utilizados en diagramas de flujo.....	18
Aplicaciones para diseño de Diagramas de Flujo.....	19
PSeInt.....	19
Raptor.....	19
SmartDraw.....	19
Bibliografía.....	20

Estructuras de Control: Parte I

Les doy la bienvenida a nuestra primera unidad del curso Programación I, en la cual les propondremos una serie de conceptos relacionados con el paradigma de programación estructurada aplicada al lenguaje de programación C++. Es de gran importancia que interioricen los conceptos que se plantearán en este módulo del curso, dado que se seguirán utilizando por el resto de su vida en el área de programación de dispositivos electrónicos, por lo que los invitamos a reflexionar sobre este tema y las aplicaciones en el mundo de la electrónica digital.

Iniciamos nuestro camino

Antes de escribir un programa que dé solución a un problema, es necesario tener una comprensión detallada de todo el problema, además de una metodología cuidadosamente planeada para resolverlo. Al escribir un programa, también debemos comprender los tipos de bloques de construcción disponibles, y emplear las técnicas comprobadas para construir programas. Los conceptos aquí presentados son imprescindibles para formar las bases de la programación estructurada de los nuevos aprendientes en el mundo de la programación, que es base fundamental en la electrónica digital tanto en el desarrollo de software como de hardware.

En esta primera parte de la unidad presentamos las instrucciones **if**, **if...else** y **while** de C++, tres de los bloques de construcción que permiten a los programadores especificar la lógica requerida para formar programas estructurados que realicen las funciones especificadas. Además, presentamos los operadores de asignación de C++ y exploramos los operadores de incremento y decremento que pueden ayudar a abreviar y simplificar muchas instrucciones de los programas.

Antes de iniciar con las estructuras de control es necesario conocer sobre un par de conceptos fundamentales cuando se es nuevo en el mundo de la programación tales como: algoritmos y pseudocódigo.

Cualquier problema de computación puede resolverse ejecutando una serie de acciones en un orden específico. Un procedimiento para resolver un problema en términos de las acciones a ejecutar y el orden en el que se ejecutan estas acciones se conoce como un algoritmo.

El pseudocódigo (o “imitación” de código) es un lenguaje informal que ayuda a los programadores a desarrollar algoritmos sin tener que preocuparse por los estrictos detalles de la sintaxis del lenguaje. El pseudocódigo es similar al lenguaje cotidiano; es conveniente y amigable con el usuario, aunque no es realmente un lenguaje de programación de computadoras. El pseudocódigo no se ejecuta en las computadoras. En lugar de ello, ayuda al programador a “organizar” un programa antes de intentar escribirlo en un lenguaje de programación como C++.

Por lo general, en un programa las instrucciones se ejecutan una después de otra, en el orden en que están escritas. Este proceso se conoce como ejecución secuencial. Varias instrucciones en C++ que pronto veremos permiten al programador especificar que la siguiente instrucción a ejecutarse tal vez no sea la siguiente en la secuencia, donde esto se conoce como transferencia de control.

C++ tiene tres tipos de instrucciones de selección las cuales se describen en esta unidad. La instrucción de selección **if** realiza una acción si la condición es verdadera, o evita la acción si la condición es falsa. La instrucción de selección **if...else** realiza una acción si la condición es verdadera, o realiza una acción distinta si la condición es falsa.

La instrucción de selección **if** es una instrucción de selección simple, ya que selecciona o ignora una sola acción o un solo grupo de acciones. La instrucción **if...else** se conoce como instrucción de selección doble, ya que selecciona entre dos acciones distintas o grupos de acciones.

C++ sólo tiene tres tipos de estructuras de control, a las cuales nos referiremos de aquí en adelante como instrucciones de control: la instrucción de secuencia, las instrucciones de selección (tres tipos: **if**, **if...else** y **switch**) y las instrucciones de repetición (tres tipos: **while**, **for** y **do...while**) las cuales veremos en la segunda parte de esta unidad. Cada programa de C++ se forma combinando tantas instrucciones de secuencia, selección y repetición como sea apropiado para el algoritmo que implemente el programa. Las instrucciones de control de una sola entrada/una sola salida, facilitan la creación de programas; donde las instrucciones de control están unidas entre sí mediante la conexión del punto de salida de una instrucción de control, al punto de entrada de la siguiente. Este procedimiento es similar a la manera en que un niño apila los bloques de construcción, así que a esto le llamamos apilamiento de instrucciones de control. Por otro lado, una manera alternativa de conectar las instrucciones de control es por medio del anidamiento de instrucciones de control, en el cual una instrucción de control aparece dentro de otra. Por lo tanto, los algoritmos en los programas en C++ se crean a partir de sólo tres tipos de instrucciones de control, que se combinan solamente de dos formas.

Instrucción de selección if

Los programas utilizan instrucciones de selección para elegir entre los cursos alternativos de acción. Por ejemplo, suponga que la calificación para aprobar un examen es de 60.

La instrucción en pseudocódigo

*Si la calificación del estudiante es mayor o igual a 60
Imprimir "Aprobado"*

La instrucción anterior **if** en pseudocódigo puede escribirse en C++ de la siguiente manera:

```
if ( calificacionEstudiante >= 60 ) {  
    cout << "Aprobado";  
}
```

El código anterior determina si la condición “*la calificación del estudiante es mayor o igual a 60*” es verdadera (**true**) o falsa (**false**). Si la condición es verdadera se imprime “*Aprobado*”, y se “ejecuta” en orden la siguiente instrucción. Si la condición es falsa se ignora la instrucción para imprimir, y se ejecuta en orden la siguiente instrucción en la línea de código. Observe que la segunda línea de esta instrucción de selección tiene sangría. Dicha sangría es opcional, pero se recomienda ya que enfatiza la estructura inherente de los programas estructurados. Al convertir pseudocódigo en código de C++, el compilador ignora los caracteres de espacio en blanco (como espacios, tabuladores y caracteres de nueva línea) que se utilizan para aplicar sangría y espaciado vertical.

La figura 1.1 muestra el diagrama de flujo de una instrucción **if** simple



Figura 1.1 Instrucción if simple

En la figura 1.1 podemos observar que si la condición es verdadera (sí) ejecuta el caso verdadero, en caso contrario la condición es falsa (no) se termina con la ejecución del programa.

Instrucción de selección if...else

La instrucción if de selección simple realiza una acción indicada solamente cuando la condición es verdadera (**true**); de no ser así, se evita dicha acción. La instrucción **if...else** de selección doble permite al programador especificar una acción a realizar cuando la condición es verdadera, y otra distinta cuando la condición es falsa (**false**). Por ejemplo, la instrucción en pseudocódigo

Si la calificación del estudiante es mayor o igual a 60

Imprimir "Aprobado"

De lo contrario

Imprimir "Reprobado"

El pseudocódigo anterior imprime "Aprobado" si la calificación del estudiante es mayor o igual a 60, e imprime "Reprobado" si la calificación del estudiante es menor a 60. En cualquier caso, después de que ocurre la impresión se "ejecuta" la siguiente instrucción en pseudocódigo en la secuencia.

La instrucción anterior **if...else** en pseudocódigo puede escribirse en C++ como

```
if ( calificacion >= 60 ) {
    cout << "Aprobado";
} else {
    cout << "Reprobado";
}
```

La figura 1.2 muestra el diagrama de flujo de una instrucción **if...else**

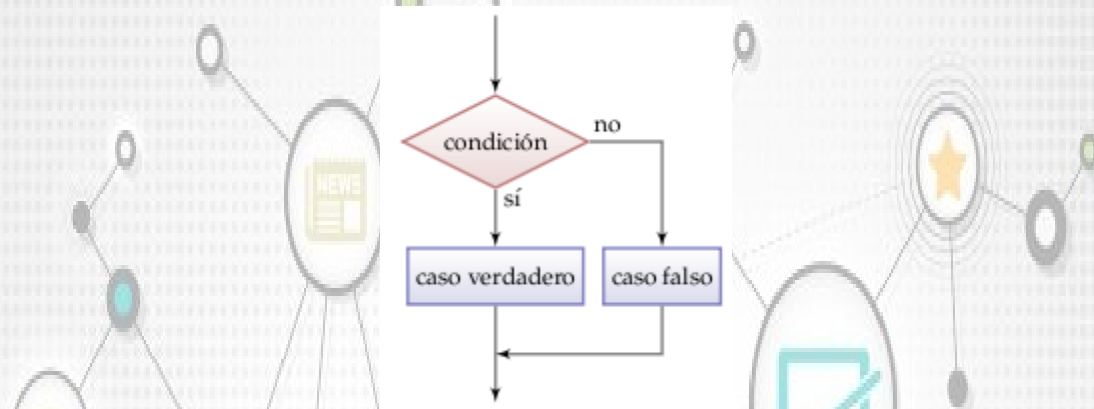


Figura 1.2 Instrucción if...else

En la figura 1.2 podemos observar que si la condición es verdadera (sí) ejecuta el caso verdadero, en caso contrario la condición es falsa (no) ejecuta el caso falso. En ambos casos una vez que se ejecuten las instrucciones de los respectivos casos finaliza la ejecución del programa.

Instrucción de repetición while

Una instrucción de repetición (también llamada instrucción de ciclo, o un ciclo) permite al programador especificar que un programa debe repetir una acción mientras cierta condición sea verdadera. La instrucción en pseudocódigo

```
Mientras existan más artículos en mi lista de compras  
  Comprar el siguiente artículo y quitarlo de mi lista
```

El pseudocódigo anterior describe la repetición que ocurre durante una salida de compras. La condición “existan más artículos en mi lista de compras” puede ser verdadera o falsa. Si es verdadera, entonces se realiza la acción “Comprar el siguiente artículo y quitarlo de mi lista”.

Esta acción se realizará en forma repetida mientras la condición sea verdadera. La instrucción contenida en la instrucción de repetición Mientras (while) constituye el cuerpo de esta estructura, el cual puede ser una sola instrucción o un bloque. En algún momento, la condición se hará falsa (cuando el último artículo de la lista de compras sea adquirido y eliminado de la lista). En este punto la repetición terminará y se ejecutará la primera instrucción que esté después de la instrucción de repetición.

La figura 1.3 muestra el diagrama de flujo de una instrucción **while**

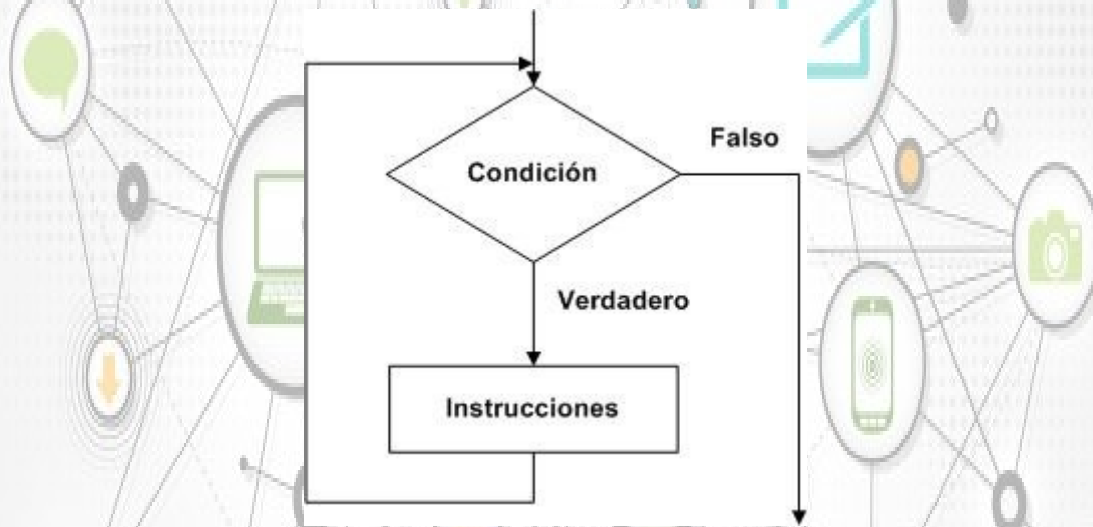


Figura 1.3 Instrucción while

Explicaremos más detalles sobre esta instrucción de repetición en la segunda parte de esta unidad.

Para finalizar, es de gran importancia tener claros los conceptos sobre estructuras de control selectivas, dado que la integración de este tipo de instrucciones es base de la programación estructurada que seguirá utilizando en su desarrollo profesional en el área digital y de software de ingeniería en electrónica, por lo que lo invitamos a poner especial atención a este tema junto con su implementación en el lenguaje de programación C++.

Estructuras de Control: Parte II

Les doy la bienvenida a la segunda parte de esta unidad del curso Programación I, en la cual les propondremos una serie de conceptos relacionados con el paradigma de programación estructurada aplicada al lenguaje de programación C++. Es de gran importancia que interiorices los conceptos que se plantearán en esta segunda parte, dado que se seguirán utilizando por el resto de su vida en el área de programación de dispositivos electrónicos, por lo que los invitamos a reflexionar sobre este tema y las aplicaciones en el mundo de la electrónica digital.

La primera parte los introdujo a los tipos de bloques de construcción disponibles para solucionar problemas, donde utilizamos dichos bloques de construcción para emplear las técnicas ya comprobadas de la construcción de programas. En esta segunda parte de la unidad continuaremos nuestra presentación de la teoría y los principios de la programación estructurada, presentando el resto de las instrucciones de control. En esta unidad demostraremos las instrucciones **for**, **while**, **do...while** y **switch**, a través de una serie de ejemplos cortos en los que utilizaremos las instrucciones **while** y **for** controladas por contador.

Instrucción de repetición while

Una instrucción de repetición (también llamada instrucción de ciclo, o un ciclo), permite al programador especificar que un programa debe repetir una acción mientras cierta condición sea verdadera. La instrucción en pseudocódigo

Mientras existan más artículos en mi lista de compras

Comprar el siguiente artículo y quitarlo de mi lista

El pseudocódigo anterior describe la repetición que ocurre durante una salida de compras. La condición “existan más artículos en mi lista de compras” puede ser verdadera o falsa. Si es verdadera, entonces se realiza la acción “Comprar el siguiente artículo y quitarlo de mi lista”.

Esta acción se realizará en forma repetida mientras la condición sea verdadera. La instrucción contenida en la instrucción de repetición Mientras (while), constituye el cuerpo de esta estructura, el cual puede ser una sola instrucción o un bloque. En algún momento, la condición se hará falsa (cuando el último artículo de la lista de compras sea adquirido y eliminado de la lista). En este punto la repetición terminará y se ejecutará la primera instrucción que esté después de la instrucción de repetición.

Un ejemplo de la instrucción de repetición while en C++ es el siguiente: considere un segmento de programa diseñado para encontrar la primera potencia de tres que sea mayor a cien. Suponga que la variable producto de tipo int se inicializa en tres. Cuando la siguiente instrucción while termine de ejecutarse, el producto contendrá el resultado:

```
int producto = 3;  
while ( producto <= 100 ) {  
    producto = 3 * producto;  
}
```

Cuando esta instrucción while comienza a ejecutarse, el valor de producto es tres. Cada repetición de la instrucción while multiplica a producto por tres, por lo que producto toma los valores de 9, 27, 81 y 243, sucesivamente. Cuando producto se vuelve 243, la condición de la instrucción while (producto <= 100) se torna falsa. Esto termina la repetición, por lo que el valor final de producto es 243. En este punto, la ejecución del programa continúa con la siguiente instrucción después de la instrucción while.

La figura 1.4 muestra el diagrama de flujo de una instrucción **while**

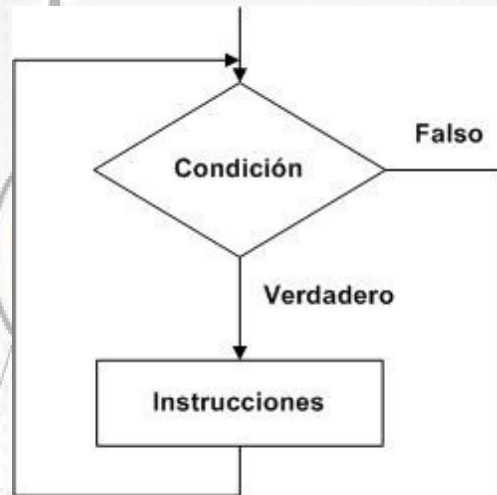


Figura 1.4 Instrucción while

Como vimos en el ejemplo anterior se utiliza la instrucción de repetición **while** para formalizar los elementos requeridos para llevar a cabo la repetición controlada por medio de un contador. Este tipo de repetición requiere

- El nombre de una variable de control (o contador de ciclo)
- El valor inicial de la variable de control
- La condición de continuación de ciclo, que evalúa el valor final de la variable de control (es decir, determina si el ciclo debe continuar o no)
- El incremento (o decremento) con el que se modifica la variable de control cada vez que pasa por el ciclo.

Instrucción de repetición for

La sección anterior presentó los aspectos necesarios para la repetición controlada por contador. La instrucción while puede utilizarse para implementar cualquier ciclo controlado por un contador. C++ también cuenta con la instrucción de repetición **for**, la cual especifica los detalles de la repetición controlada por contador en una sola línea de código.

La figura 1.5 muestra los principales componentes de una instrucción **for** con su respectiva explicación de cada una de sus partes.

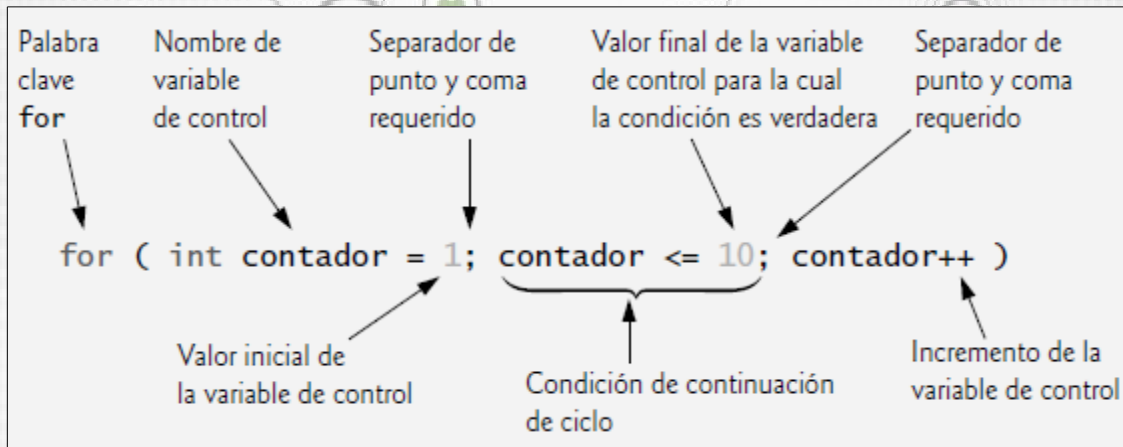


Figura 1.5 Componentes del encabezado de la instrucción **for**

Observe que el encabezado de la instrucción **for** “se encarga de todo”: especifica cada uno de los elementos necesarios para la repetición controlada por contador con una variable de control. Si hay más de una instrucción en el cuerpo del **for**, se requieren llaves para encerrar el cuerpo del ciclo.

La figura 1.6 muestra el diagrama de flujo de una instrucción **for**

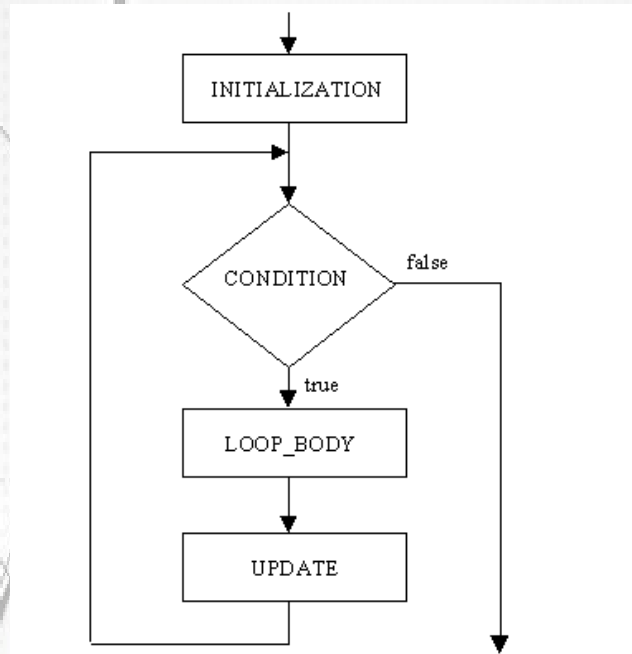


Figura 1.6 Instrucción for

Un ejemplo del uso de la instrucción **for** se muestra en la figura 1.7

```

1 // Suma de enteros con la instrucción for.
2 #include <iostream>
3
4 using namespace std;
5
6 int main(){
7     int total = 0; // inicializa el total
8
9     // obtiene el total de los enteros pares del 2 al 20
10    for ( int numero = 2; numero <= 20; numero += 2 ){
11        total += numero;
12    }
13    cout << "La suma es " << total << endl; // muestra los resultados
14    return 0; // terminó correctamente
15 } // fin de main
    
```

Figura 1.7 Suma de enteros con ciclo for

Instrucción de repetición do...while

La instrucción **do...while** es similar a la instrucción **while**. En la instrucción **while**, la evaluación de la condición de continuación de ciclo ocurre al principio del ciclo, antes de ejecutar su cuerpo. La instrucción **do...while** evalúa la condición de continuación de ciclo después de ejecutar el cuerpo del ciclo; por lo tanto, el cuerpo del ciclo siempre se ejecutará cuando menos una vez. Cuando termina una instrucción **do...while**, la ejecución continúa con la instrucción que va después de la cláusula **while**. No es necesario utilizar llaves en la instrucción **do...while** si sólo hay una instrucción en el cuerpo; sin embargo, la mayoría de los programadores incluyen las llaves para evitar la confusión entre las instrucciones **while** y **do...while**.

La figura 1.8 muestra el diagrama de flujo de una instrucción **do...while**

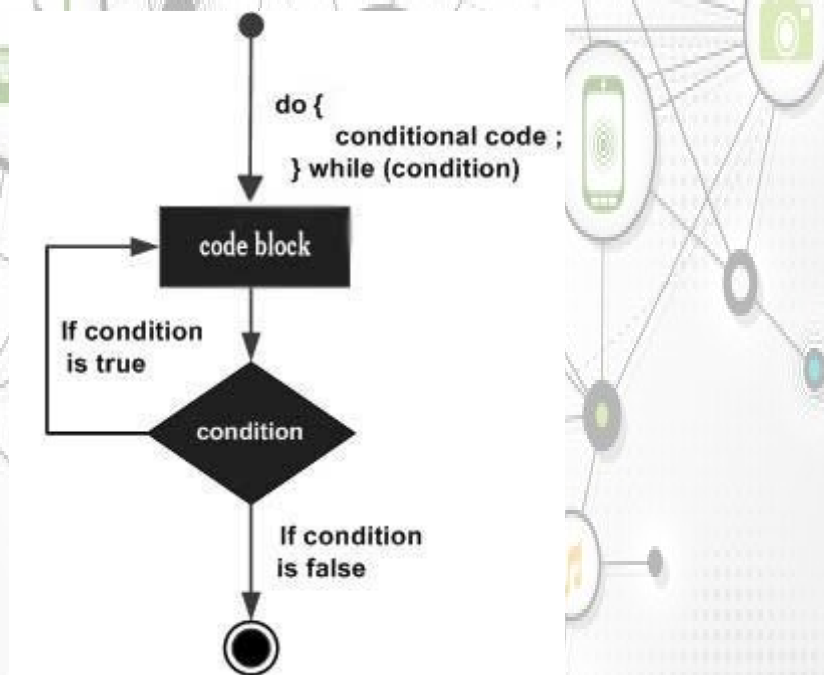


Figura 1.8 Instrucción do...while

Un ejemplo del uso de la instrucción **do...while** se muestra en la figura 1.9

```

1 // La instrucción de repetición do...while.
2
3 #include <iostream>
4
5 using namespace std;
6
7 int main(){
8
9     int contador = 1; // inicializa contador
10
11     do{
12         cout << contador << " "; // muestra contador
13         contador++; // incrementa contador
14     } while ( contador <= 10 ); // fin de do...while
15
16     cout << endl; // imprime una nueva línea
17     return 0; // indica que terminó correctamente
18 } // fin de main

```

Figura 1.9 Impresión números del 1 al 10 con do...while

La figura 1.9 utiliza una instrucción **do...while** para imprimir los números del uno al diez. Al entrar a la instrucción **do...while**, en la línea 12 se imprime el valor de contador y en la línea 13 se incrementa contador. Después el programa evalúa la prueba de continuación de ciclo al final del mismo (línea 14). Si la condición es verdadera, el ciclo continúa a partir de la primera instrucción del cuerpo en la instrucción **do...while** (línea 12). Si la condición es falsa, el ciclo termina y el programa continúa con la siguiente instrucción después del ciclo (línea 16).

Es de gran importancia tener claros los conceptos sobre estructuras de control selectivas y repetitivas, dado que la integración de estos dos tipos de instrucciones son la base de la programación estructurada que seguirá utilizando en su desarrollo profesional en el área digital y de software de ingeniería en electrónica, por lo que lo invitamos a poner especial atención a este tema junto con su implementación en el lenguaje de programación C++.

Diagramas de Flujo

Un diagrama de flujo es una representación visual del tratamiento de un problema; como mínimo, un diagrama de flujo muestra las operaciones realizadas dentro del proceso que pretende la solución del problema y la secuencia en la que son realizadas dichas operaciones. A grandes rasgos, un diagrama de flujo debe responder a las siguientes preguntas: ¿Qué actividades debo hacer para solucionar el problema? ¿En qué orden deberán ejecutarse dichas actividades?

Los diagramas de flujo generalmente son elaborados en las fases iniciales del ciclo de vida de desarrollo (análisis, diseño), dado que facilitan la comunicación entre los desarrolladores y los clientes; una de sus mayores virtudes es que facilitan el entendimiento de la lógica del problema a resolver, transformando problemas grandes y complejos en una serie ordenada de actividades por realizar.



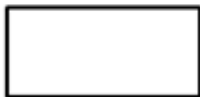


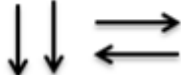
Ventajas de utilizar diagramas de flujo

Las ventajas de utilizar diagramas de flujo son las siguientes:

1. Son generalmente conocidos e interpretados.
2. Son una de las mejores formas de comunicar la lógica de un programa o sistema.
3. Permiten el análisis de un problema de una forma abstracta.
4. Son aceptados como documentación técnica para los programas y desarrollos.
5. Sirven como referencia para los programadores.
6. Apoyan el proceso de depuración de programas, sobre todo en la solución de problemas relacionados con la lógica del programa, mismos que el compilador es incapaz de detectar.
7. Facilitan las funciones de mantenimiento a los programas, al exponerlos de una forma clara y lógica.

Símbolos utilizados en diagramas de flujo

Los siguientes símbolos son algunos de los utilizados en el desarrollo de un diagrama de flujo de una aplicación o programa de software.

Simbología	Pseudocódigo	Función
	Inicio o Fin	Se utiliza para empezar y terminar un programa
	Conocer o pedir	Utilizar para pedir datos por teclado estos pueden ser números, textos o datos alfanuméricos
	Evaluar o Asignación	Sirve para evaluar operaciones aritméticas como formular y también para asignar.
	Preguntas de decisión	Sirve para hacer decisiones.
	Imprimir	Imprime el Resultado
	Flechas de Dirección	Sirven para guardar la dirección del flujo del programa

Como material de lectura complementario pueden consultar el siguiente documento de la aplicación de software SmartDraw <https://www.smartdraw.com/flowchart/simbolos-de-diagramas-de-flujo.htm>.

Aplicaciones para diseño de Diagramas de Flujo

Algunas aplicaciones para el diseño, implementación y simulación de diagramas de flujo se listan a continuación:

PSeInt

PSeInt es una herramienta para asistir a un estudiante en sus primeros pasos en programación. Mediante un simple e intuitivo pseudolenguaje en español (complementado con un editor de diagramas de flujo), le permite centrar su atención en los conceptos fundamentales de la algoritmia computacional, minimizando las dificultades propias de un lenguaje y proporcionando un entorno de trabajo con numerosas ayudas y recursos didácticos.

URL: <http://pseint.sourceforge.net/>

Raptor

Raptor Diagramas de Flujo es un entorno de programación basado en diagramas de flujo, diseñado específicamente para ayudar a los estudiantes de programación o a los propios programadores profesionales a visualizar sus algoritmos y evitar algunas de las complicaciones propias de la sintáctica de programación.

URL: <https://raptor-diagramas-de-flujo.programas-gratis.net/>

SmartDraw

SmartDraw es una herramienta de diagrama utilizada para hacer diagramas de flujo, organigramas, mapas mentales, diagramas de proyectos y otros elementos visuales de negocios. SmartDraw tiene dos versiones: una edición en línea y una edición descargable para escritorio de Windows.

URL: <https://www.smartdraw.com/flowchart/diagramas-de-flujo.htm>

Bibliografía

Deitel & Deitel. (2010). *Programación intermedia con Java*. Pearson Prentice Hall. México.

Dale, N. (2007). *Programación y resolución de problemas con C++*. México: McGraw Hill.

Galeano, G. (2009). *Programación de Sistemas Embebidos en C*. Colombia: Alfaomega.

García-Bermejo, J. (2008). *Programación Estructurada en C*. Madrid, España: Pearson Educación.

Largo, E. (2016). Las cinco fases para resolver un problema de programación. Recuperado de <https://www.ecodeup.com/como-resolver-problemas-programacion/>

Márquez, G. (2011). *Introducción a la Programación Estructurada en C*. México: Pearson Educación.

Savitch, Walter. (2007). *Resolución de problemas en C++*. México: Pearson Educación.

Wilson, R. (2013). Introducción a la *Programación en C++*. Recuperado de <http://rubenprogramacionenc.blogspot.com/2013/05/resumen-cap-4.html>

Conclusiones

- Se transformó el entorno presencial del curso Programación I a una modalidad virtual, lo que ayudará a mejorar el problema de ausentismo, abandono del curso por temas laborales y disponibilidad de tiempo, al contar con un entorno virtual de aprendizaje.
- Se implementó el entorno virtual de aprendizaje para el curso Programación I, que ayudará a los estudiantes a tener acceso a todos los contenidos del curso, evaluaciones y actividades desde el inicio, ayudando a administrar mejor el tiempo y acceder al curso desde cualquier lugar, en el momento que el estudiante tenga la disponibilidad para su desarrollo.
- Se capacitó a un docente en entornos virtuales de aprendizaje, lo que ayudará a mejorar el proceso de incorporación de tecnologías virtuales en la carrera de ingeniería electrónica en los cursos que se ajusten para la implementación de un entorno virtual de aprendizaje.
- Se migró el cien por ciento de la evaluación del curso a un entorno virtual, lo que ayudará al estudiante a tener la flexibilidad en realizar las asignaciones desde un lugar remoto donde se encuentre ubicado y con flexibilidad de horario.